Jaroslav Pokorny · Vaclav Repa
Karel Richta · Wita Wojtkowski
Henry Linger · Chris Barry
Michael Lang *Editors*

Information Systems Development

Business Systems and Services: Modeling and Development

Springer

Information Systems Development

Jaroslav Pokorny • Vaclav Repa • Karel Richta
Wita Wojtkowski • Henry Linger • Chris Barry
Michael Lang
Editors

Information Systems Development

Business Systems and Services: Modeling and Development

Springer

Editors
Dr. Jaroslav Pokorny
Charles University Praha
Faculty of Mathematics & Physics
Department of Software Engineering
Malostranske nam. 25
118 00 Praha 1, Czech Republic
pokorny@ksi.ms.mff.cuni.cz

Dr. Vaclav Repa
Prague University of Economics
Department of Information Technologies
Attn. Prof. Jiri Vorisek
Nam. W. Churchilla 4
130 67 Praha
Czech Republic
repa@vse.cz

Dr. Karel Richta
Charles University, Praha
Faculty of Mathematics & Physics
Malostranske nam. 25
118 00 Praha 1
Czech Republic
richta@ksi.mff.cuni.cz

Wita Wojtkowski
Boise State University
College of Business & Economics
Department of Information Technology
& Supply Chain Management
1910 University Dr.
Boise, Idaho 83725-1615, USA
riswoj2t@cobfac.boise.state.edu

Dr. Henry Linger
Monash University
Faculty of Information Technology
Clayton Victoria
Australia
Henry.Linger@infotech.monash.edu.au

Ph.D. Chris Barry
National University of Ireland
Department of Accountancy & Finance
University Road
Galway, Ireland

Michael Lang
National University of Ireland
Department of Accountancy & Finance
University Road
Galway, Ireland
michael.lang@nuigalway.ie

ISBN 978-1-4419-9645-9 e-ISBN 978-1-4419-9790-6
DOI 10.1007/978-1-4419-9790-6
Springer New York Dordrecht Heidelberg London

Library of Congress Control Number: 2011933916

Printed on acid-free paper

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

Preface

This book is the outcome the 19th International Conference on Information Systems Development (ISD 2010), hosted by the faculty of Mathematics and Physics, Charles University in Prague during 25–27 August 2010.

The ISD conference evolved from the first Polish-Scandinavian Seminar on Current Trends in Information Systems Development Methodologies held in Poland in 1988. It was a great honor for us to organize the nineteenth within this series of conferences.

The development of information systems is still one of main theoretical and practical disciplines usable in all areas of society, including government, industry, science, community, and in the home. The theme for the ISD 2010 Conference was Business Systems and Services: Modeling and Development. The Conference program was scheduled into the following sessions: Business Process Management (BPM), Business Systems Analysis & Design I.–II, Model-Driven Engineering in ISD, Agile and High-Speed Systems Development, IS/IT Project Management I.–II, Information System Development Methodology I.–III, Web Services and SOA in the BPM Context I.–II, Cognitive Aspects of Business Systems and Services, Public Information Systems Development I.–II, Data and Information Systems Modeling I.–III, Service Oriented Modeling, Managing of IS Development, and Miscellaneous Aspects of ISD.

In addition, there were three presentations of ISD 2010 keynote speakers: Jan L.G. Dietz (Delft University of Technology, The Netherlands), John Traxler (University of Wolverhampton, United Kingdom), and Robert Winter (University of St. Allen, Switzerland).

This book contains all papers scheduled for presentation at ISD 2010 Conference, except those contributions accepted as posters. EasyChair tool was used as conference software both for authors and Program Committee members. In response to the Call for Papers, 112 were received by the submission deadline of April 25, 2010. Of them, 62 were selected by the Program Committee for presentation. This represents an acceptance rate approximately 55%. All papers were peer reviewed by 3 or 4 referees.

We have also selected two papers to share the Best Paper Award. The first one is “Adapting the Lesk Algorithm for Calculating Term Similarity in the Context of

Requirements Engineering" by Jürgen Vöhringer and Günther Fliedl. The second one is "Towards understanding contradictions in enterprise system implementations—insights from a case study" by Stig Nordheim.

We are grateful to the international advisory committee for providing guidance and moral support during the submission process. We also thank to those behind the scene, i.e. to members of the Program Committee for their timely evaluation of submitted papers. Then we would also like to thank and acknowledge the work of Alena Buchalcevová for managing EasyChair system and Matúš Ondreička for valuable activities with preparation and maintenance of the conference web site. We are also grateful to Action M Agency for its effort in local arrangements.

Prague, August 2010

Jaroslav Pokorný, General Chair
Karel Richta and Václav Řepa, PC Chairs
Matúš Ondreička, Proceedings Editor

Conference Organization

The 19th International Conference on Information Systems Development was hosted by Faculty of Mathematics and Physics, Charles University in Prague, and organized by Department of Software Engineering in cooperation with Action M Agency, Prague. The theme of the conference is "Business Systems and Services: Modelling and Development". The organization and management of such a major international conference requires the collaboration and dedication of very many people. We are especially grateful to our international programme committee who voluntarily gave their time to review the submissions. The excellent standard of papers contained within this volume bears testimony to the diligence and rigour of the peer review process. We are also very appreciative of the efforts of all the committee.

General Chair

Jaroslav Pokorný	Charles Univeristy, Prague, Czech Republic

Programme Chairs

Karel Richta	Charles Univeristy, Prague, Czech Republic
Václav Řepa	University of Economics, Prague, Czech Republic

International Advisory Committee

Chris Barry	National University of Ireland, Galway, Ireland
Michael Lang	National University of Ireland, Galway, Ireland
Henry Linger	Monash University, Melbourne, Australia
Wita Wojtkowski	Boise State University, Boise, ID, USA

Organising Committee Chair

Milena Ziethamlová	Action M Agency, Prague, Czech Republic

Organising Committee

Alena Buchalcevová	University of Economics, Prague, Czech Republic
Pavla Kozáková	Action M Agency, Prague, Czech Republic
Jiří Novák	Charles Univeristy, Prague, Czech Republic
Matúš Ondreička	Charles Univeristy, Prague, Czech Republic
Helena Palovská	University of Economics, Prague, Czech Republic
Blanka Puková	Action M Agency, Prague, Czech Republic

International Program Committee

Backlund Per	University of Skovde, Sweden
Barry Chris	National University of Ireland, Galway, Ireland
Barzdins Janis	University of Latvia, Riga, Latvia
Borzovs Juris	Information Technology Institute, Latvia
Buchalcevová Alena	University of Economics, Prague, Czech Republic
Burdescu Dumitru	University of Craiova, Romania
Butleris Rimantas	Kaunas University of Technology, Lithuania
Caplinskas Albertas	Institute of Mathematics and Informatics, Vilnius, Lithuania
Connor Andy	Auckland University of Technology, New Zealand
Dalcher Darren	Middlesex University, London, United Kingdom
Du Xiaofeng	British Telecom, United Kingdom
Eessaar Erki	Tallinn University of Technology, Estonia
Eriksson Owen	Dalarna University College, Borlnge, Sweden
Escalona María-José	University of Sevilla, Spain
Feuerlicht George	University of Technology Sydney, Australia
Fredriksson Odd	Karlstad University, Sweden
Fisher Julie	Monash University, Melbourne, Australia
Guizzardi Giancarlo	Federal University of Espirito Santo, Vitoria, Brazil
Granić Andrina	University of Split, Croatia
Grundspenkis Janis	Riga Technical University, Latvia
Gustas Remigijus	Karlstad University, Sweden
Haav Hele-Mai	Institute of Cybernetics at Tallinn Technical University, Estonia
Hawryszkiewycz Igor	University of Technology Sydney, Australia
Hevner Alan	University of South Florida, USA
Ivanov Sergey	George Washington University, USA

Ivanovic Mirjana	University of Novi Sad, Serbia and Montenegro
Jaccheri Letizia	Norwegian University of Science and Technology, Norway
Kautz Karlheinz	Copenhagen Business School, Denmark
Kirikova Marite	Riga Technical University, Latvia
Krishna Aneesh	Curtin University of Technology, Perth, Australia
Kuusik Rein	Tallinn University of Technology, Estonia
Kuznetsow Sergei	Institute for System Programming of Russian Academy of Science, Russia
Lang Michael	National University of Ireland, Galway, Ireland
Leppanen Mauri	University of Jyvaskyla, Finland
Linger Henry	Monash University, Australia
Maciaszek Leszek A.	Macquarie University, Australia
Magyar Gabor	Magyar Budapest University of Technology and Economics, Hungary
Moreton Robert	University of Wolverhampton, United Kingdom
Nemuraite Lina	Kaunas Technical University, Lithuania
Nilsson Anders G.	Karlstad University, Sweden
Noran Ovidiu	Griffith University, Australia
Norbjerg Jacob	Copenhagen Business School, Denmark
Palovská Helena	University of Economics, Prague, Czech Republic
Pankowska Malgorzata	University of Economics in Katowice, Poland
Paspallis Nearchos	University of Cyprus, Cyprus
Pastor Oscar	University of Valencia, Spain
Pirotte Alain	University of Louvain, Belgium
Plantak Vukovac Dijana	University of Zagreb, Croatia
Ramos Isidro	Universitat Politecnica de Valencia, Spain
Riha Karel	Kingston University, London, United Kingdom
Rouibah Kamel	Kuwait University, Kuwait
Schiopiu Burlea Adriana	University of Craiova, Romania
Söderström Eva	University of Skovde, Sweden
Sokolovsky Zbynek	Technical University in Darmstadt, Germany
Song William	Durham University, United Kingdom
Stamelos Ioannis	Aristotle University, Greece
Strahringer Susanne	Technische Universitat Dresden, Germany
Sukovskis Uldis	Riga Technical University, Latvia
Traxler John	University of Wolverhampton, United Kingdom
Unold Jacek	Wroclaw University of Economics, Poland
Vasilecas Olegas	Vilnius Gediminas Technical University, Lithuania
Vossen Gottfried	University of Munster, Germany
Welch Christine	University of Porthmouth, United Kingdom
Winter Robert	University St. Gallen, Switzerland
Wrycza Stanislaw	University of Gdansk, Poland
Zendulka Jaroslav	Brno University of Technology, Czech Republic
Zurada Jozef	University of Louisville, USA
Zupančič Jože	University od Maribor, Slovenia

Referees

Per Backlund
Chris Barry
Janis Barzdins
Juris Borzovs
Dumitru Burdescu
Adriana Schiopiu Burlea
Rimantas Butleris
Albertas Caplinskas
Andy Connor
Darren Dalcher
Xiaofeng Du
Erki Eessaar
Owen Eriksson
María-José Escalona
George Feuerlicht
Julie Fisher
Odd Fredriksson
Andrina Granić
Janis Grundspenkis
Giancarlo Guizzardi
Remigijus Gustas
Hele-Mai Haav
Igor Hawryszkiewycz
Alan Hevner
Sergey Ivanov
Mirjana Ivanovic
Letizia Jaccheri
Karlheinz Kautz
Marite Kirikova
Aneesh Krishna
Rein Kuusik
Sergei Kuznetsow
Michael Lang
Mauri Leppanen
Henry Linger
Leszek A. Maciaszek
Gabor Magyar
Robert Moreton
Lina Nemuraite
Anders G. Nilsson
Ovidiu Noran
Jacob Norbjerg
Matúš Ondreička
Malgorzata Pankowska
Nearchos Paspallis
Oscar Pastor
Alain Pirotte
Isidro Ramos
Karel Riha
Kamel Rouibah
Zbynek Sokolovsky
William Song
Ioannis Stamelos
Susanne Strahringer
Uldis Sukovskis
Eva Söderström
John Traxler
Jacek Unold
Olegas Vasilecas
Gottfried Vossen
Dijana Plantak Vukovac
Christine Welch
Robert Winter
Stanislaw Wrycza
Jaroslav Zendulka
Jože Zupančič
Jozef Zurada

External Reviewers

Aijaz Soomro
Alessander Benevides
Algirdas Laukaitis
Anna-Sofia Alklind
Taylor
Audris Kalnins
Deniss Kumlander
Diana Kalibatiene
Edgars Rencis
Gunnar Thies
Gytenis Mikulenas
Jens Lechtenbörger
Jesper Holgersson
Kamel Rouibah
Karin Sjöholm
Panagiotis Katsaros
Sergejs Kozlovics
Sergejus Sosunovas
Tarmo Veskioja

Sponsoring Institutions

The conference ISD 2010 was organized under the Auspices of the Mayor of the City of Prague, Pavel Bém.

Contents

List of Contributors

Anas Alsoud Informatics Research Centre, University of Reading, Whiteknights, Reading RG6 6WB, UK
e-mail: a.alsoud@student.reading.ac.uk

Saleh Al-wahaishi VŠB–Technical University of Ostrava, Ostrava-Poruba, Czech Republic
e-mail: salehw@vsb.cz

Peter Oluoch Ating'a Department of Computing, Curtin University of Technology, Perth, Australia
e-mail: p.atinga@student.curtin.edu.au

Irene Barba Universidad de Sevilla, Sevilla, Spain
e-mail: irenebr@us.es

Chris Barry National University of Ireland, Galway, Ireland
e-mail: chris.barry@nuigalway.ie

Peter Bartalos Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies, Slovak University of Technology, Ilkovičova 3, 842 16 Bratislava, Slovakia
e-mail: bartalos@fiit.stuba.sk

Tomàš Bartoš Department of Software Engineering, Malostranské naměstí 25, 118 00 Prague 1, Czech Republic
e-mail: bartos@ksi.mff.cuni.cz

Peter M. Bednar School of Computing, University of Portsmouth, Portsmouth, UK
e-mail: peter.bednar@ics.lu.se

Department of Informatics, Lund University, Lund, Sweden

Linda Bergkvist Department of Information Systems & Project Management, Karlstad University, Karlstad, Sweden
e-mail: linda.bergkvist@kau.se

Ghassan Beydoun University of Wollongong, Wollongong, Australia
e-mail: beydoun@uow.edu.au

Mària Bielikovà Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies, Slovak University of Technology, Ilkovičova 3, 842 16 Bratislava, Slovakia
e-mail: bielik@fiit.stuba.sk

Geoffrey Black 1910 University Drive, Boise, ID, USA
e-mail: gblack@boisestate.edu

Narasimha Bolloju Department of Information Systems, City University of Hong Kong, Hong Kong, China
e-mail: narsi.bolloju@cityu.edu.hk

Vicente Botti Universidad Politecnica de Valencia, Valencia, Spain
e-mail: vbotti@dsic.upv.es

Lukas Burkon Department of Information Technology, University of Economics, Prague, W. Churchill Sq. 4, Prague, Czech Republic
e-mail: lukas.burkon@vse.cz

Frada Burstein Faculty of Information Technology, Monash University, Melbourne, Australia
e-mail: frada.burstein@infotech.monash.edu.au

David W. Bustard University of Ulster, Coleraine, Northern Ireland
e-mail: dw.bustard@ulster.ac.uk

Niall Byrne KTL, Nass, Ireland
e-mail: niallbyrne@ktl.ie

Tomas Cerny Department of Computer Science and Engineering, Czech Technical University, Charles square 13, 121 35 Prague 2, Czech Republic
e-mail: tomas.cerny@fel.cvut.cz

Changxuan Wan School of Information Technology, Jiangxi University of Finance and Economics, Nanchang, China

Jiangxi Key Laboratory of Data and Knowledge Engineering, Jiangxi University of Finance & Economics, Nanchang, China

Emma Chavez-Mora Bond University, Robina, QLD, Australia

Universidad Catolica de la Ssma. Concepcion, Concepcion, Chile
e-mail: emchavez@bond.edu.au

Kieran Conboy Business Information Systems, J.E. Cairnes School of Business & Economics, National University of Ireland, Galway, Ireland
e-mail: kieran.conboy@nuigalway.ie

Kevin Curran University of Ulster, Londonderry, Northern Ireland
e-mail: kj.curran@ulster.ac.uk

Christian Czarnecki Detecon International GmbH, Oberkasseler Str. 2, 53227, Bonn, Germany

Faculty of Computer Science, Otto-von-Guericke-University, Universitaetsplatz 2, 39106 Magdeburg, Germany
e-mail: christian.czarnecki@detecon.com

Darren Dalcher National Center for Project Management, Middlesex University, Trent Park, London, N14 4YZ, UK
e-mail: D.Dalcher@mdx.ac.uk

Carmelo Del Valle Universidad de Sevilla, Sevilla, Spain
e-mail: carmelo@us.es

Jan L.G. Dietz Delft University of Technology, P.O. Box 5031, The Netherlands, 2600 GA Delft, Dalian 116025 China
e-mail: j.l.g.dietz@tudelft.nl

Gunnar Dietz Global Institute of Management and Economics, Dongbei University of Finance and Economics, 116025, Dalian, China
e-mail: mail@gdietz.de

Francisco José Dominguez-Mayo Department of Computer Languages and Systems, University of Seville, Seville, Spain
e-mail: fjdominguez@us.es

Michael J. Donahoo Department of Computer Science, Baylor University, P.O. Box 97356, 76798-7356, Waco, TX, USA
e-mail: jeff donahoo@baylor.edu

Lynette Drevin North-West University, Private Bag X6001, Potchefstroom, South Africa, 2520
e-mail: Lynette.Drevin@nwu.ac.za

Maria Jose Escalona Department of Computer Languages and Systems, University of Seville, Seville, Spain
e-mail: mjescalona@us.es

George Feuerlicht Department of Information Technology, University of Economics, Prague, W. Churchill Sq. 4, Prague, Czech Republic
e-mail: jirif@vse.cz

Faculty of Engineering and Information Technology, University of Technology, Sydney, P.O. Box 123 Broadway, Sydney, NSW 2007, Australia
e-mail: george.feuerlicht@uts.edu.au

Gavin Finnie Bond University, Robina, QLD, Australia
e-mail: gfinnie@bond.edu.au

Günther Fliedl Institute of Applied Informatics, Alpen-Adria-Universität Klagenfurt, Klagenfurt, Austria
e-mail: guenther.fliedl@ifit.uni-klu.ac.at

Kam Hay Fung School of Information Systems, Technology and Management, The University of New South Wales, Kensington, NSW, Australia
e-mail: kamhayfung@ieee.org

Emilia Garcia Universidad Politecnica de Valencia, Valencia, Spain
e-mail: mgarcia@dsic.upv.es

Aditya K. Ghose Decision Systems Lab., School of Computer Science and Software Engineering, University of Wollongong, Wollongong, NSW 2522, Australia
e-mail: aditya@uow.edu.au

Adriana Giret Universidad Politecnica de Valencia, Valencia, Spain
e-mail: agiret@dsic.upv.es

Abel Gómez Departamento de Sistemas Informáticos y Computación, Universitat Politècnica de València, Cno. de Vera, s/n. Valencia, Spain
e-mail: agomez@dsic.upv.es

Andreas Gregoriades Department of Computer Science and Engineering, European University, Nicosia, Cyprus
e-mail: a.gregoriades@eu.cy

Guoqiong Liao School of Information Technology, Jiangxi University of Finance and Economics, Nanchang, China

Jiangxi Key Laboratory of Data and Knowledge Engineering, Jiangxi University of Finance & Economics, Nanchang, China

Remigijus Gustas Department of Information Systems, Karlstad University, Karlstad, Sweden
e-mail: remigijus.gustas@kau.se

Prima Gustiené Department of Information Systems, Karlstad University, Karlstad, Sweden
e-mail: prima.gustiene@kau.se

Till Haselmann European Research Center for Information Systems (ERCIS), WWU Münster, Münster, Germany
e-mail: haselmann@wi.uni-muenster.de

Nicklas Holmberg Department of Informatics, Lund School of Economics and Management, Lund University, Lund, Sweden
e-mail: nicklas.holmberg@ics.lu.se

Janis Iljins University of Latvia, Raina bulv. 19, 1586 Riga, Latvia
e-mail: janis.iljins@lu.lv

Ahmad Jaffar UAE University, Al-Ain, United Arab Emirates
e-mail: ajaffar@uaeu.ac.ae

Martin Juhrisch Chair for Information Systems, Dresden University of Technology, 01062, Dresden, Germany
e-mail: martin.juhrisch@tu-dresden.de

Diana Kalibatiene Vilnius Gediminas Technical University, Sauletekio al. 11, 10223, Vilnius, Lithuania
e-mail: diana@isl.vgtu.lt

Dimka Karastoyanova Institute of Architecture of Application Systems, University of Stuttgart, Universitätsstraße 38, 70569 Stuttgart, Germany
e-mail: dimka.karastoyanova@iaas.uni-stuttgart.de

Jàn Kasarda Department of Software Engineering, Malostranské naměstí 25, 118 00 Prague 1, Czech Republic
e-mail: jan.kasarda@mff.cuni.cz

Marite Kirikova Riga Technical University, Riga, Latvia
e-mail: marite.kirikova@cs.rtu.lv

Andràs Ferenc Kovàcs Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, Budapest, Hungary
e-mail: kovacsaf@gmail.com

Natalija Kozmina Faculty of Computing, University of Latvia, Riga LV-1586, Latvia
e-mail: natalija.kozmina@lu.lv

Jaroslav Král Charles University, Malostranské nàm. 25, 118 00, Prague, Czech Republic
e-mail: kral@ksi.mff.cuni.cz

Aneesh Krishna Department of Computing, Faculty of Science and Engineering, Curtin University of Technology, Perth, Australia
e-mail: a.krishna@curtin.edu.au

Tri A. Kurniawan Decision Systems Lab., School of Computer Science and Software Engineering, University of Wollongong, Wollongong, NSW 2522, Australia
e-mail: tak976@uow.edu.au

Richard Lai Department of Computer Science and Computer Engineering, La Trobe University, Melbourne, Australia
e-mail: lai@cs.latrobe.edu.au

Juha Lamminen Department of Computer Science and Information Systems, University of Jyväskylä, P.O. Box 35 (Agora), 40014, Jyväskylä, Finland
e-mail: juha.e.lamminen@jyu.fi

Lesley P. W. Land School of Information Systems, Technology and Management, The University of New South Wales, Sydney, NSW 2052, Australia
e-mail: l.land@unsw.edu.au

Michael Lang Business Information Systems, J.E. Cairnes School of Business & Economics, National University of Ireland, Galway, Ireland
e-mail: michael.lang@nuigalway.ie

Lardner Mark National University of Ireland, Galway, Ireland
e-mail: m.lardner1@nuigalway.ie

Even Åby Larsen University of Agder, Kristiansand, Norway
e-mail: even.larsen@uia.no

Lam-Son Lě Decision Systems Lab., School of Computer Science and Software Engineering, University of Wollongong, Wollongong, NSW 2522, Australia
e-mail: lle@uow.edu.au

Lei Shu School of Information Technology, Jiangxi University of Finance and Economics, Nanchang, China

Mauri Leppänen Department of Computer Science and Information Systems, University of Jyväskylä, P.O. Box 35 (Agora), 40014, Jyväskylä, Finland
e-mail: mauri@cs.jyu.fi

Katrina Leyking IDS Scheer AG, Altenkesseler Straße 17, 66115 Saarbrücken, Germany
e-mail: katrina.leyking@ids-scheer.com

Frank Leymann Institute of Architecture of Application Systems, University of Stuttgart, Universitätsstraße 38, 70569 Stuttgart, Germany
e-mail: leymann@iaas.uni-stuttgart.de

Henry Linger Faculty of Information Technology, Monash University, Melbourne, Australia
e-mail: henry.linger@infotech.monash.edu.au

Garry Lohan National University of Ireland, Galway, Ireland
e-mail: Garry.Lohan@nuigalway.ie

Antonio A. Lopez-Lorca University of Wollongong, Wollongong, Australia
e-mail: aall645@uowmail.edu.au

Graham C. Low School of Information Systems, Technology and Management, The University of New South Wales, Kensington, NSW, Australia
e-mail: g.low@unsw.edu.au

Gàbor Magyar Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, Budapest, Hungary
e-mail: magyar@tmit.bme.hu

Bartosz Marcinkowski Department of Business Informatics, University of Gdansk, Sopot, Poland
e-mail: bartosz.marcinkowski@gmail.com

Vojtech Mates Faculty of Information Technology, Brno University of Technology, Brno, Czech Republic
e-mail: imates@fit.vutbr.cz

Helena McCabe KTL, Naas, Ireland
e-mail: Lena.McCabe@ktl.ie

Orla McHugh Business Information Systems, J.E. Cairnes School of Business & Economics, National University of Ireland, Galway, Ireland
e-mail: orla.mchugh@nuigalway.ie

Yuan Miao School of Engineering and Science, Victoria University, Melbourne, 3011, Australia
e-mail: yuan.miao@vu.edu.au

Tim Miller University of Melbourne, Melbourne, Australia
e-mail: tmiller@unimelb.edu.au

Martin Molhanec Czech Technical University in Prague, Technickà 2, 166 27, Praha 6, Czech Republic
e-mail: molhanec@fel.cvut.cz

Juan Manuel Moreno-Rivera SIGTel Geomàtica, Centro I+D de Empresas and Grupo de Teledetección y SIG, Instituto de Desarrollo Regional, University of Castilla-La Mancha, Albacete, Spain
e-mail: juanmanuel.moreno@sigtel.es

Cleopa John Mushi School of Systems Engineering, University of Reading, Whiteknights, Reading, RG6 6AY, UK
e-mail: c.j.s.k.mushi@reading.ac.uk

Elena Navarro Computing Systems Department, University of Castilla-La Mancha, Albacete, Spain
e-mail: enavarro@info-ab.uclm.es

Laila Niedrite Faculty of Computing, University of Latvia, Raina bulv. 19, LV-1586, Riga, Latvia
e-mail: Laila.Niedrite@lu.lv

Aivars Niedritis University of Latvia, Riga, Latvia
e-mail: Aivars.Niedritis@lu.lv

Anders G. Nilsson Karlstad University, SE-651 88, Karlstad, Sweden
e-mail: anders.nilsson@kau.se

Stig Nordheim University of Agder, Postboks 422, 4604, Kristiansand, Norway
e-mail: stig.nordheim@uia.no

Tihomir Orehovački Faculty of Organization and Informatics, University of Zagreb, Varaždin, Croatia
e-mail: tihomir.orehovacki@foi.hr

Tero Päivärinta University of Agder, Kristiansand, Norway
e-mail: tero.paivarinta@uia.no

Vicente Pelechano Universidad Politécnica de Valencia, Valencia, Spain
e-mail: pele@pros.upv.es

Tomàš Pitner Faculty of Informatics, Masaryk University, Brno, Czech Republic
e-mail: tomp@fi.muni.cz

Jaroslav Pokorný Charles University of Prague, Prague, Czech Republic
e-mail: pokorny@ksi.mff.cuni.cz

Hamidreza Pousti Faculty of Information Technology, Monash University, Melbourne, Australia
e-mail: Hamid.Pousti@infotech.monash.edu.au

Jaroslav Prochazka Tieto/University of Ostrava, Ostrava, Czech RePublic
e-mail: Jaroslav Prochazka@osu.cz

Isidro Ramos Departamento de Sistemas Informáticos y Computación, Universitat Politècnica de València, Cno. de Vera, s/n. Valencia, Spain
e-mail: iramos@dsic.upv.es

Vaclav Repa Department of Information Technology, University of Economics, W. Churchill sqr. 4, 130 67, Prague 3, Czech Republic
e-mail: repa@vse.cz

Manuel Mejías Risoto Department of Computer Languages and Systems, University of Seville, Seville, Spain
e-mail: risoto@us.es

Marek Rychlý Department of Information Systems, Faculty of Information Technology, Brno University of Technology, Božetěchova 2, 612 66, Brno, Czech Republic
e-mail: rychly@fit.vutbr.cz

Pertti Saariluoma Department of Computer Science and Information Systems, University of Jyväskylä, P.O. Box 35 (Agora), 40014, Jyväskylä, Finland
e-mail: pertti.saariluoma@jyu.fi

Paula Savolainen School of Computing, University of Eastern Finland, 1627, 70211, Kuopio, Finland

Lero–The Irish Software Engineering Research Centre, University of Limerick, Ireland
e-mail: paula.m.savolainen@uef.fi

Christoph Schneider Department of Information Systems, City University of Hong Kong, Hong Kong, China
e-mail: christoph.schneider@cityu.edu.hk

David Schumm Institute of Architecture of Application Systems, University of Stuttgart, Universitätsstraße 38, 70569, Stuttgart, Germany
e-mail: david.schumm@iaas.uni-stuttgart.de

Estefanía Serral Universidad Politécnica de Valencia, Valencia, Spain
e-mail: eserral@pros.upv.es

Jaroslav Škrabálek Faculty of Informatics, Masaryk University, Brno, Czech Republic
e-mail: skrabalek@fi.muni.cz

Jiří Slabý Faculty of Informatics, Masaryk University, Brno, Czech Republic
e-mail: xslaby@fi.muni.cz

Václav Slavíček Faculty of Informatics and Management, University of Hradec Kralové, Rokitanského 62, 500 03, Hradec Kralové, Czech Republic
e-mail: vaclav.slavicek@uhk.cz

Kari Smolander Lappeenranta University of Technology, Lappeenranta, Finland
e-mail: kari.smolander@lut.fi

Václav Snášel VŠB–Technical University of Ostrava, Ostrava-Poruba, Czech Republic
e-mail: vaclav.snasel@vsb.cz

Piotr Soja Department of Computer Science, Cracow University of Economics, Rakowicka 27, 31-510, Kraków, Poland
e-mail: eisoja@cyf-kr.edu.pl

Darja Solodovnikova University of Latvia, Raina bulv. 19, LV-1586, Riga, Latvia
e-mail: darja.solodovnikova@lu.lv

William Wei Song School of Engineering and Computing Sciences, Durham University, Durham, United Kingdom
e-mail: w.w.song@durham.ac.uk

Myra Spiliopoulou Faculty of Computer Science, Otto-von-Guericke-University, Universitaetsplatz 2, 39106, Magdeburg, Germany
e-mail: myra@iti.cs.uni-magdeburg.de

Odd Steen Department of Informatics, Lund School of Economics and Management, Lund University, Lund, Sweden
e-mail: odd.steen@ics.lu.se

Leon Sterling University of Melbourne, Melbourne, Australia
e-mail: leon@cs.mu.oz.au

Swinburne University of Technology, Healesville, Australia
e-mail: lsterling@groupwise.swin.edu.au

Steve Strauch Institute of Architecture of Application Systems, University of Stuttgart, Universitätsstraße 38, 70569, Stuttgart, Germany
e-mail: strauch@iaas.unistuttgart.de

Renate Strazdina Riga Technical University, Riga, Latvia
e-mail: renate.strazdina@cs.rtu.lv

Lily Sun School of Systems Engineering, University of Reading, Whiteknights, Reading, RG6 6AY, UK
e-mail: lily.sun@reading.ac.uk

Gábor Szűcs Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, Budapest, Hungary
e-mail: szucs@tmit.bme.hu

Meredith A. Taylor University of Bern, 6 Hallerstrasse, Bern, Switzerland
e-mail: meredithtaylor@boisestate.edu

Lucia Tokárová Faculty of Informatics, Masaryk University, Brno, Czech Republic
e-mail: tokarov@fi.muni.cz

Jesús Torres Department of Computer Languages and Systems, University of Seville, Seville, Spain
e-mail: jtorres@us.es

John Traxler Learning Lab, University of Wolverhamton, TF2 9NT, Telford, UK
e-mail: john.ttraxler@wlv.ac.uk

Maris Treimanis University of Latvia, Raina bulv. 19, 1586, Riga, Latvia
e-mail: maris.treimanis@lu.lv

Azmat Ullah Department of Computer Science and Computer Engineering, La Trobe University, Melbourne, Australia
e-mail: Ullah@latrobe.edu.au

Pedro Valderas Universidad Politécnica de Valencia, Valencia, Spain
e-mail: pvalderas@pros.upv.es

Olegas Vasilecas Vilnius Gediminas Technical University, Sauletekio al. 11, 10223, Vilnius, Lithuania
e-mail: olegas@isl.vgtu.lt

June M. Verner School of Computer Science and Engineering, The University of New South Wales, Sydney, NSW 2052, Australia
e-mail: june.verner@gmail.com

Jürgen Vöhringer Institute of Applied Informatics, Alpen-Adria-Universität Klagenfurt, Klagenfurt, Austria
e-mail: juergen.voehringer@ifit.uni-klu.ac.at

Ivo Vondrák VŠB–Technical University of Ostrava, Ostrava-Poruba, Czech Republic
e-mail: ivo.vondrak@vsb.cz

Gottfried Vossen European Research Center for Information Systems (ERCIS), WWU Münster, Münster, Germany
e-mail: vossen@uni-muenster.de

Christine Welch Department of Strategy and Business Systems, University of Portsmouth, Portsmouth, UK
e-mail: christine.welch@port.ac.uk

Axel Winkelmann European Research Center for Information Systems (ERCIS), Westfaelische Wilhelms-University, Leonardo-Campus 3, 48149, Muenster, Germany
e-mail: winkelmann@ercis.uni-muenster.de

Robert Winter Institute of Information Management, University of St. Gallen, Müller-Friedberg-Strasse 8, 9000, St. Gallen, Switzerland
e-mail: robert.winter@unisg.ch

Wita Wojtkowski 1910 University Drive, Boise, ID, USA
e-mail: wwojtkow@boisestate.edu

Stanislaw Wrycza Department of Business Informatics, University of Gdansk, Sopot, Poland
e-mail: swrycza@univ.gda.pl

Xuebing Yang School of Engineering and Science, Victoria University, Melbourne, 3011, Australia
e-mail: xuebing.yang@live.vu.edu.au

Michal Žemliçka Charles University, Malostranské nàm. 25, 118 00, Prague, Czech Republic
e-mail: zemlicka@ksi.mff.cuni.cz

Yanchun Zhang School of Engineering and Science, Victoria University, Melbourne, 3011, Australia
e-mail: Yanchun.Zhang@vu.edu.au

From Information Systems Development to Enterprise Engineering

Jan L. G. Dietz

Abstract For decades already, the field of information systems development is facing challenges for which it is not equipped. Examples are business process management, service oriented design, and model-driven engineering. The only way out is to broaden the scope from information system to enterprise. Enterprise Engineering is an emerging discipline, originating from both the Information System Sciences and the Organizational Sciences. This means a major paradigm shift. On the paradigmatic basis that has been developed up to now, the notions of Enterprise Ontology, Enterprise Architecture, and Enterprise Governance are firmly founded. Next to broadening the scope towards enterprise engineering, there is an urgent need for scientific rigor. A first step in that direction is provided by the combination of three theories: the ϕ-theory, the τ-theory, and the ψ-theory. Together they are able to ban the bullshit from the discipline of information system engineering and to prevent it to creep into the discipline of enterprise engineering.

1 Introduction

The complexity of (automated) information systems (ICT applications) has become unmanageable. The ICT applications landscape in every major enterprise comprises thousands of applications, which overlap each other to a large extent, but which also leave substantial parts of the enterprise's organization unsupported. Moreover, ICT applications still do not meet an enterprise's organization requirements and/or its business expectations, and there is still no correlation between the ICT investments of an enterprise and its revenues [21]. Decisions to (re-)develop ICT applications are strategic initiatives. Unfortunately, the vast majority of strategic initiatives fail, meaning that enterprises are unable to gain success from their strategy [7]. The high failure rates are reported from various domains: total quality management, business process reengineering, six sigma, lean production, e-business, customer

J. L. G. Dietz (✉)
Delft University of Technology, P.O. Box 5031, 2600 GA, Delft, The Netherlands
e-mail: j.l.g.dietz@tudelft.nl

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_1,

relationship management, as well as from mergers and acquisitions. It appears that these failures are mostly the avoidable result of an inadequate implementation of the strategy. Rarely are they the inevitable consequence of a poor strategy. Abundant research indicates that the key reason for strategic failures is the lack of coherence and consistency, collectively also called congruence, among the various components of an enterprise[1]. At the same time, the need to operate as an integrated whole is becoming increasingly important [14, 18].

Globalization, the removal of trade barriers, deregulation, etc., have led to networks of cooperating enterprises on a large scale, enabled by the virtually unlimited possibilities of modern information and communication technology. Future enterprises will therefore have to operate in an ever more dynamic and global environment. They need to be more agile, more adaptive, and more transparent. In addition, they will be held more publicly accountable for every effect they produce.

Focusing on the development of ICT[2] applications, two major causes can be identified of the current disastrous situation. The first one is the insufficient level of professionalism of ICT practitioners (information architects, business process architects, enterprise architects, etc.). This regards in particular their understanding of organizations and business processes. Despite the availability of appropriate knowledge, information systems professionals stick to so-called best practices, which lack any scientific justification. Theoretically solid knowledge and practically useful methodologies based on it, are brushed aside for allegedly being too complicated or "not invented here". It is interesting to draw the analogy with an engineering discipline that is only some 50 years older, namely aeronautical engineering. Imagine that aeronautical engineers, in designing and building an aircraft, would ignore the laws of aerodynamics for being too complicated, and would refuse to consider demonstrably better materials and building techniques for being "not invented here". Obviously, such a practice would immediately lead to broad societal protests and to the fall of governments. Not so for the engineering discipline of applying ICT in enterprises. There are no societal protests and governments do not fall. Apparently, society accepts the frequent malfunctioning of vital systems, the irritating non-compliancy of ICT applications with human needs and expectations, and the continuous huge waste of money.

One might argue that other people than ICT professionals, notably organizational professionals, should bring in the appropriate knowledge of organizations and business processes in an ICT project. Unfortunately, however, they also lack appropriate knowledge, which constitutes the second major cause of the current disastrous situation. Organizational professionals address organizational problems with black box thinking based knowledge, i.e. knowledge concerning the function and the behavior of enterprises, as dominantly present in the organizational sciences. Such knowledge is sufficient, and perfectly adequate, for managing an enterprise. However, it is definitely inadequate for changing an enterprise's organization (which is what one

[1] By an enterprise is understood any collaboration of people for achieving a common goal. It ranges from the organization of a birthday party to a corporate company or a governmental ministry.

[2] ICT stands for Information and Communication Technology.

does when developing information systems). In order to bring about these changes, white-box based knowledge is needed, i.e. knowledge concerning the construction and the operation of enterprises. Developing and applying such knowledge requires no less than a paradigm shift in our thinking about enterprises.

2 The Needed Paradigm Shift

The current situation in the organizational sciences resembles very much the one that existed in the information systems sciences around 1970. At that time, a revolution took place in the way people conceived information technology and its applications. The pioneer of this revolution was Langefors [15]. Since then, people have been aware of the distinction between the *form* and the *content* of information (cf. Fig. 2). This revolution marks the transition from the era of Data Systems Engineering to the era of Information Systems Engineering. The comparison we draw with the information systems sciences is not an arbitrary one. On the one hand, the key enabling technology for shaping future enterprises is the modern information and communication technology (ICT). On the other hand, there is a growing insight in the information systems sciences that the central notion for understanding profoundly the relationship between organization and ICT is the entering into and complying with commitments between social individuals. This insight has been brought to us by the Language Action Perspective (LAP) movement in the 1990s of the twentieth century [4]. Figure 1 exhibits the LAP-based analysis of a communicative act (a term adopted from Habermas [10]).

Commitments are raised in communication, through the so-called *intention* of communicative acts. Examples of intentions are requesting, promising, stating, and accepting. Therefore, as the content of communication was put on top of its form in the 1970s, the intention of communication is now put on top of its content. It explains and clarifies the organizational notions collaboration and cooperation, as well as authority and responsibility. It also puts organizations definitely in the category of social systems, very distinct from information systems. Said revolution in the information systems sciences marks the transition from the era of Information Systems Engineering to the era of Enterprise Engineering, while at the same time merging with relevant parts of the Organizational Sciences, as illustrated in Fig. 2.

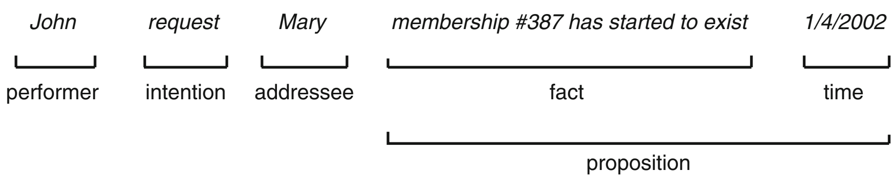


Fig. 1 General structure of a communicative act

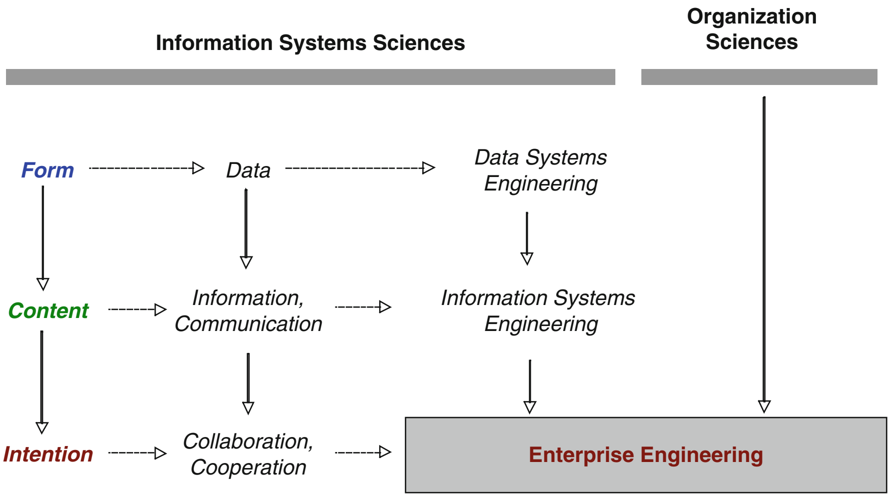


Fig. 2 The evolution of enterprise engineering

The mission of the discipline of Enterprise Engineering[3] is to combine (relevant parts from) the organizational sciences and the information systems sciences, and to develop theories and methodologies for the analysis, design, and implementation of future enterprises. Two crucial concepts have already emerged that are considered paramount for accomplishing this mission: Enterprise Ontology and Enterprise Architecture. A precondition for incorporating these methodologies effectively in an enterprise is the being well established of Enterprise Governance.

Theoretically, *Enterprise Ontology* is the understanding of an enterprise's construction and operation in a fully implementation independent way. Practically, it is the highest-level constructional model of an enterprise, the implementation model being the lowest one [5]. Compared to its implementation model, the ontological model offers a reduction of complexity of well over 90%. Only by applying this notion of Enterprise Ontology can substantial changes of enterprises be made intellectually manageable. Theoretically, *Enterprise Architecture* is the normative restriction of design freedom. Practically, it is a coherent and consistent set of principles that guide the (re)design and (re)implementation of an enterprise, and that come in addition to the specific requirements in a change project [11]. These principles are derived from the enterprise's strategic basis (mission, vision). Only by applying this notion of Enterprise Architecture can consistency be achieved between the strategic basis and the operational business rules of an enterprise. *Enterprise Governance* is the organizational competence for continuously exercising guiding authority over enterprise strategy and architecture development, and the subsequent design, implementation, and operation of the enterprise [12]. Adopting this notion of enterprise

[3] See www.ciaonetwork.org.

governance enables an enterprise to be compliant with external and internal rules, and to perform in an optimal and societally responsible way.

3 The Enterprise Engineering Paradigm

Addressing the challenges mentioned above requires a paradigm shift. As said before, it is the mission of the discipline of Enterprise Engineering to develop new, appropriate theories, models, methods and other artifacts for the analysis, design, implementation, and governance of enterprises by combining (relevant parts of) management and organization science, information systems science, and computer science. The ambition is to address (all) traditional topics in said disciplines from the *Enterprise Engineering Paradigm*, whose objective is to make enterprise engineering intellectually manageable. Let us summarize the three major motivations:

1. The vast majority of strategic initiatives fail, meaning that enterprises are unable to gain success from their strategy [13]. Abundant research indicates that the key reason for strategic failures is the lack of coherence and consistency among the various components of an enterprise. At the same time, the need to operate as a unified and integrated whole is becoming increasingly important [14, 18]. These challenges are dominantly addressed from a functional or managerial perspective, as advocated by management and organization science. Such knowledge is necessary and sufficient for managing an enterprise, but it is inadequate for bringing about changes. To do that, one needs to take a constructional or engineering perspective.
2. Both organizations and software systems are complex and prone to entropy. This means that in the course of time, the costs of bringing about similar changes increase in a way that is known as combinatorial explosion [16, 17]. Regarding (automated) information systems, this has been demonstrated; regarding organizations, it is still a conjecture. Entropy can be reduced and managed effectively through modular design based on atomic elements.
3. The people in an enterprise are collectively responsible for the operation (including management) of the enterprise. In addition, they are collectively responsible for the evolution of the enterprise (adapting to needs for change). These responsibilities can only be borne if the people in the enterprise have appropriate (ontological) knowledge of the enterprise [5].

The Enterprise Engineering Paradigm is expressed in seven postulates, which are listed and discussed hereafter. These postulates have been formulated by the CIAO! Network, which is an international community of researchers and practitioners in enterprise engineering. They are published by the CIAO! Network as the Enterprise Engineering Manifesto[4].

[4] See www.ciaonetwork.org.

3.1 Postulate 1

In order to perform optimally and to implement changes successfully, enterprises must operate as a unified and integrated whole. *Unity* and *integration* can only be achieved through deliberate enterprise development (including design, engineering, and implementation) and governance.

3.2 Postulate 2

Enterprises are essentially *social systems*, of which the elements are human beings in their role of social individuals, bestowed with appropriate authority and bearing the corresponding responsibility. The operating principle of enterprises is that these human beings enter into and comply with commitments regarding the products/services that they create/deliver. These coordination acts occur, together with the concerned production act, in universal patterns called transactions.

Note Human beings may be supported by technical artifacts of all kinds, notably by ICT systems. Therefore, enterprises are often referred to as socio-technical systems. However, only human beings are responsible and accountable for what the supporting technical artifacts do.

3.3 Postulate 3

There are two distinct perspectives on enterprises (as on all systems): *function* and *construction*. All other perspectives are a subdivision of one of these. Accordingly, there are two distinct kinds of models: black-box models and white-box models. White-box models are objective; they regard the construction of a system. Black-box models are subjective; they regard a function of a system, which is a relationship between the system and some stakeholder(s). Both perspectives are needed for developing enterprises.

Note For convenience sake, we talk about the business of an enterprise when taking the function perspective of the customer, and about its organization when taking the construction perspective.

3.4 Postulate 4

In order to manage the complexity of a system (and to reduce and manage its entropy), one must start the constructional design of the system with its ontological model, which is a fully implementation independent model of the construction and the operation of the system. Moreover, an ontological model has a modular structure and its elements are (ontologically) atomic.

For enterprises the meta model of such models is called *enterprise ontology*. For information systems the meta model is called information system ontology.

Note At any moment in the life time of a system, there is only one ontological model, capturing its actual construction, though abstracted from its implementation. The ontological model of a system is comprehensive and concise, and extremely stable.

3.5 Postulate 5

It is an ethical necessity for bestowing *authority* on the people in an enterprise, and having them bear the corresponding *responsibility*, that these people are able to internalize the (relevant parts of the) ontological model of the enterprise, and to constantly validate the correspondence of the model with the operational reality.

Note It is a duty of enterprise engineers to provide the means to the people in an enterprise to internalize its ontological model.

3.6 Postulate 6

To ensure that an enterprise operates in compliance with its strategic concerns, these concerns must be transformed into generic functional and constructional normative principles, which guide the (re-)development of the enterprise, in addition to the applicable specific requirements. A coherent, consistent, and hierarchically ordered set of such principles for a particular class of systems is called an architecture. The collective architectures of an enterprise are called its *enterprise architecture*.

Note The term “architecture” is often used (also) for a model that is the outcome of a design process, during which some architecture is applied. We do not recommend this homonymous use of the word.

3.7 Postulate 7

For achieving and maintaining unity and integration in the (re-)development and operation of an enterprise, organizational measures are needed, collectively called governance. The organizational competence to take and apply these measures on a continuous basis is called *enterprise governance*.

4 The Need for Scientific Rigor

Next to the paradigm shift, as discussed in Sect. 3, there is an urgent need for more scientific rigor in the information systems sciences. We have illustrated this point in Sect. 1 on the basis of an analogy with aeronautical engineering. The first call for more rigor in our discipline has been articulated during the famous NATO Conference on software engineering in 1968[5]. Notably Edsger Dijkstra has always put emphasis on it [8]. He taught about the important difference between constructing an abstract machine (a mathematical automaton) that would solve a problem and the implementation of that machine in a suitable programming language, and that designing software can only become intellectually manageable if these three techniques are well understood and applied: *separation of concerns, effective use of abstraction,* and *devising appropriate concepts*. Actually, these intellectual techniques apply to the design of all kinds of systems, including information systems and organizations.

The disastrous current situation in the area of applying ICT to organizations can very well be qualified by the term "bullshit", thereby referring to the serious philosophical study by Frankfurt [9]. Let me quote from his book:

"Bullshit is unavoidable whenever circumstances require someone to talk without knowing what he is talking about." Alas, such circumstances are omnipresent in modern society. Typical bullshitters are consultants (of all kinds), since it is of vital importance for them to appear knowledgeable in the eyes of the client. Unfortunately, many practitioners and researchers in the discipline of information systems development, also belong to the class of bullshitters. In addition, most of them seem to be unaware of their incompetence. This may look like an excuse, but actually it makes things worse, as Frankfurt teaches us:

"Bullshitters retreat from searching for correctness to personal sincerity. Since they cannot be true to the facts, they try to be true to themselves. This is the core of bullshit". Many contemporary practitioners as well as academics really believe that it is sufficient to be honest to yourself and to others, "to do your best", as one often can hear them defend their professional malperformance.

It is rather naive, of course, to think that bullshit can ever be exterminated. Somehow, people seem to need it, even to like it, judging by its luxuriant flowering all over the place. But, it can and must be banished from any serious profession, like ours. Seeking for personal sincerity instead of objective truth and correctness in research and education, or in the practice of advising enterprises and developing ICT systems, is no less than a societal crime.

In the next three sections, three theories are summarized that exemplify the scientific rigor that is needed in the discipline of enterprise engineering. They are collectively referred to by the Greek letters "$\phi\tau\psi$".

[5] http://en.wikipedia.org/wiki/Software_engineering.

5 Summary of the ϕ-Theory

Since the 1960s of the twentieth century people try to apply ICT in enterprises, for the benefit of all stakeholders. Initially, this field of human endeavor was called Electronic Data Processing (EDP). Its pursuit was to mimic the way human beings handle data by means of computers. In the 1970s, the focus shifted from the form of data to their content (cf. Fig. 1). EDP became Information Processing, performed by (automated) Information Systems (IS). It meant a paradigm shift for the IS professionals. Now the pursuit was to mimic the cognitive abilities of human beings. But what does the new paradigm exactly look like? What is the difference between data and information? What is a fact? What is an entity? In this section, the ϕ-theory is summarized[6] (the Greek letter "ϕ" is pronounced as "FI", which are the first letters of "Fact" and "Information"). It provides full and deep answers to these key questions, and many more. It is a theory about the conceptualization of factual knowledge. The ϕ-theory is rooted in semiotics [21], in ontology [1] (including mereology [20]), and in logic [22].

5.1 The Semiotic Triangle

The *semiotic triangle* portrays the important differentiation between a thought (concept) that is in the mind of a subject (i.e. a human being), the word or symbol (sign) that designates the concept and that he or she uses to communicate with others, and the thing (object) that the concept refers to and that the sign denotes. Figure 3 exhibits the semiotic triangle.

A *sign* is an object that is used as a representation of a concept, e.g. a knot in a handkerchief. Symbolic signs are structures in physical substrates, which serve

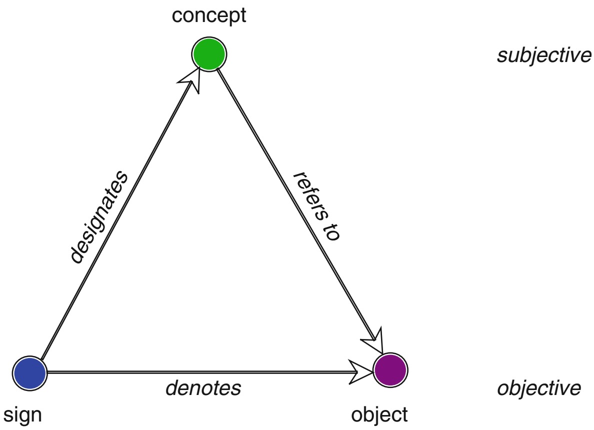


Fig. 3 The semiotic triangle

[6] For an extensive discussion of the ϕ-theory the reader is referred to [5].

to carry the signs. For example, the sign “John loves Mary” can be recorded with chalk on a blackboard, and by marks on an optical disk. An *object* is an identifiable thing. We distinguish between concrete objects (e.g. a car) and abstract objects (e.g. a membership). A *concept* is a thought or mental picture of an object. Next to these, there are “pure” concepts, which do not refer to objects. These are *scale values* in various dimensions (length, mass, time, money etc.).

Information is the inseparable duality of a sign and the concept it designates. The sign is called the *form* and the concept is called the *content* of information. *Data* are information items. Data processing (or information processing) by ICT artifacts consists of meaning preserving transformations of signs. It is a misconception that ICT artifacts, like computers, would “understand” information. Their “intelligence” is really only artificial.

5.2 The Ontological Parallelogram

The semiotic triangle does not only apply to individual concepts, but also to generic ones. In natural languages, individual concepts are typically designated by proper names (e.g. “Ludwig Wittgenstein”), and generic concepts by nouns (e.g. “person”).

A generic concept is called a *type* and a generic object is called a *class*. A class is the *extension* of a type, and a type is the *intension* of a class. The notion of type can best be understood as prescription of form, where form is some collection of (relevant) properties. If an object *conforms* to the prescription of form of a type, then its referencing concept is said to be an *instance of* the type. Next, an object is said to be a *member of* the class that is the extension of the type to which it conforms (cf. Fig. 4).

In conceptual modeling, an individual concept is called a *fact*. If an object O conforms to a type T, a fact F exists, which is an instance of T. For example, there exists the fact that Ludwig Wittgenstein is a person. Facts may also refer to multiple objects, often called aggregates. For example, the fact that Ludwig Wittgenstein is the author of the book titled “Tractatus Logico-Philosophicus” refers to a person and a book title. Each of them has its own *role* in the fact. Depending on the number of referred objects, we speak of unary, binary etc. facts. Accordingly, we speak of unary, binary, etc. types. Unary facts are also called *entities*.

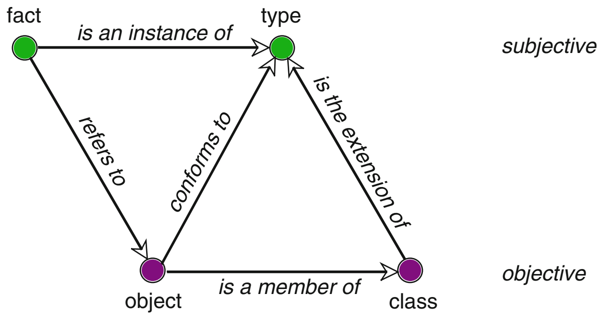


Fig. 4 The ontological parallelogram

Figure 4 exhibits the so-called *ontological parallelogram*, which is achieved by combining the semiotic triangle for individual concepts and the one for generic concepts. Signs are left out because they are ontologically irrelevant. This also avoids that signs and facts are confused (like mistaking the notation of a date in some calendar for the date it designates, which was the major cause of the Y2K problem).

The existence of a fact F1 starts from the moment that its object O conforms to type T1 and ends at the moment it ceases to do. If O also conforms to T2, there is also a fact F2 (which is an instance of T2). For example: (the object of) a person may conform to the type student and to the type patient. Note that a fact may come into existence when the form of an object changes and when a new type is defined.

5.3 Conceptual Modeling

The ontological parallelogram serves as the basis for building the *conceptual model* of a *world*. At any point in time, a world is in some state. A *state* is determined by the set of facts that exist at that moment. A state change or *transition* consists of one or more facts starting or ending to exist. The occurrence of a transition at some moment is called an *event* (therefore transitions are also called event types). Events are caused by acts in the system that the world is associated with. Therefore, events always concern *existentially independent* facts. Examples of independent facts are the person Ludwig Wittgenstein and the book titled "Tractatus Logico-Philosophicus". All other facts are *existentially dependent* on some independent fact. For example, the fact that the author of the book is Ludwig Wittgenstein is existentially dependent on the fact of the book title. It comes into existence at the moment the book title comes into existence.

Many entities have a *life cycle*: they start to exist at some moment and end to exist at another moment. In between their objects may conform to several *phase types* (like student and patient and adult).

The conceptual model of a world specifies its *state space* (i.e. the set of lawful states) and its *transition space* (i.e. the set of lawful sequences of transitions). The state space of a world is specified by means of the *type base* (i.e. the set of defined types) and the *existence laws*. These laws determine the inclusion or exclusion of the coexistence of facts. The transition space of a world is specified by the *transition base* (i.e. the set of independent fact types) and the *occurrence laws*. These determine which orders of events are mandatory and which ones are prohibited. Both the state space and the transition space can be completely specified by means of (modal) logical formulas, in practice mostly represented graphically.

5.4 Violation of the ϕ-Theory

The ontological parallelogram is the fundament of pure conceptual modeling. Among other things, it tells you that an object is a concept of some type only because the “form” of the object conforms to the type. Therefore it is erroneous to say e.g. “let us consider this thing as a system”. That is as unintelligent as saying “let us consider this professor as a parrot”. The practical experiences with DEMO[7] in building world ontologies has demonstrated the value of the ontological parallelogram. Unfortunately, a lot more examples can be provided of violations of the ϕ-theory. Many of them are actually provoked by current modeling techniques and tools. For example, almost all programming languages have the type “string”, next to “integer” and “real”. However, string is a sign type; only integer and real are concept types. Almost all data modeling techniques, even the best ones, i.e. the fact-oriented techniques like NIAM and ORM, provoke making similar errors, by allowing sign types (lexical types) in a conceptual model.

However, these fact-oriented methods do have the very important property of encouraging modelers to verify the correctness of a model by means of exemplification (instantiation) of the fact types. In particular the so-called representative population of a fact type, i.e. a set of instances that contains all basic variants, is a vital means to determine and/or verify all possible constraints. Not fact-oriented methods, like the Entity Relationship Model (ER) and all of its derivatives, e.g. the UML class model, thus the vast majority of current data modeling methods in practice do not provide this crucial help.

Let the next experience of the author serve as an illustrating example. For several years in a row, the author has been reviewer of information modeling exams of the Open University in The Netherlands, before they were held. There has been no exam without errors. This is particularly alarming since man-weeks of professional lecturers were spent on each exam. I always discovered the errors within one hour, by simply instantiating the ER models that were provided as the solutions. Moreover, when using ER one always has to normalize the resulting model in order to make sure that it contains o anomalies. In contrast, fact-oriented methods by nature result in 5th normal form models. To worsen the case, the presence of serious errors in the exams of the Open University was not an exception. Among the numerous conceptual models that I have set my eyes on, few were of acceptable quality, most suffered from various, and serious, deficiencies. Therefore we have to conclude that non-fact-oriented methods are not $\phi\tau\psi$ but bullshit.

6 Summary of the τ-Theory

In the 1990s of the twentieth century one came to realize that the ultimate goal of applying ICT should be to improve the operation of enterprises and to let ICT artifacts be an integral part of their organization. A growing need for agility of enterprises supported this idea. To achieve the goal, however, one needs a thorough understanding of systems and of the way they can be integrated.

[7] DEMO stands for Design and Engineering Methodology of Organizations (www.demo.nl).

In this section, the τ-theory is summarized[8] (the Greek letter "τ" is pronounced as "TAO", which are the first letters of "Technology", "Architecture", and "Ontology"). It is a theory about the development (i.e. design, engineering, and implementation) of artifacts of all kinds. It helps to pursue the said goal by providing deep answers to key questions like: What is the difference between function and construction? How does this distinction affect the design process? What is Technology, Architecture, and Ontology? What role do they play in system development? The τ-theory is rooted in systemics [1], and in ontology [1] (including mereology [20]).

6.1 System

The *teleological system* notion is concerned with the function and the (external) behavior of a system. It is adequate for the purpose of *using* or *controlling* a system. The *ontological system* notion is concerned with the construction and the (internal) operation of a system. It is adequate for the purpose of *building* or *changing* a system. The ontological system notion is defined as follows: something is a system if and only if it has the following properties:

- *Composition*: a set of elements that are atomic in some category (physical, social, etc.).
- *Environment*: a set of elements of the same category; the composition and the environment are disjoint.
- *Production*: the elements in the composition produce things (for the benefit of the elements in the environment).
- *Structure*: a set of influencing bonds among the elements in the composition, and between them and the elements in the environment. We distinguish between active influencing, called *interaction*, and passive influencing, called *interstriction*.

The elements in the composition and the environment are atomic with respect to the system category. For example, the elements of a social system are social individuals, or subjects (human beings). The effect of the production of a system is conceived as state changes of the system's world, as is the effect of interaction. Interstriction consists of inspecting and taking into account the state of the system's world by the elements in the composition when being active.

6.2 Model

A *white-box* (WB) model is a direct conceptualization of the ontological system definition. It represents the *construction perspective* on systems. A *black-box* (BB) model represents the *function perspective* on systems; it is actually identical to the teleological system notion. In Fig. 5, the white-box model (left) and the black-box model (right) of a car are exhibited.

---

[8] For an extensive discussion of the ϕ-theory the reader is referred to [7].

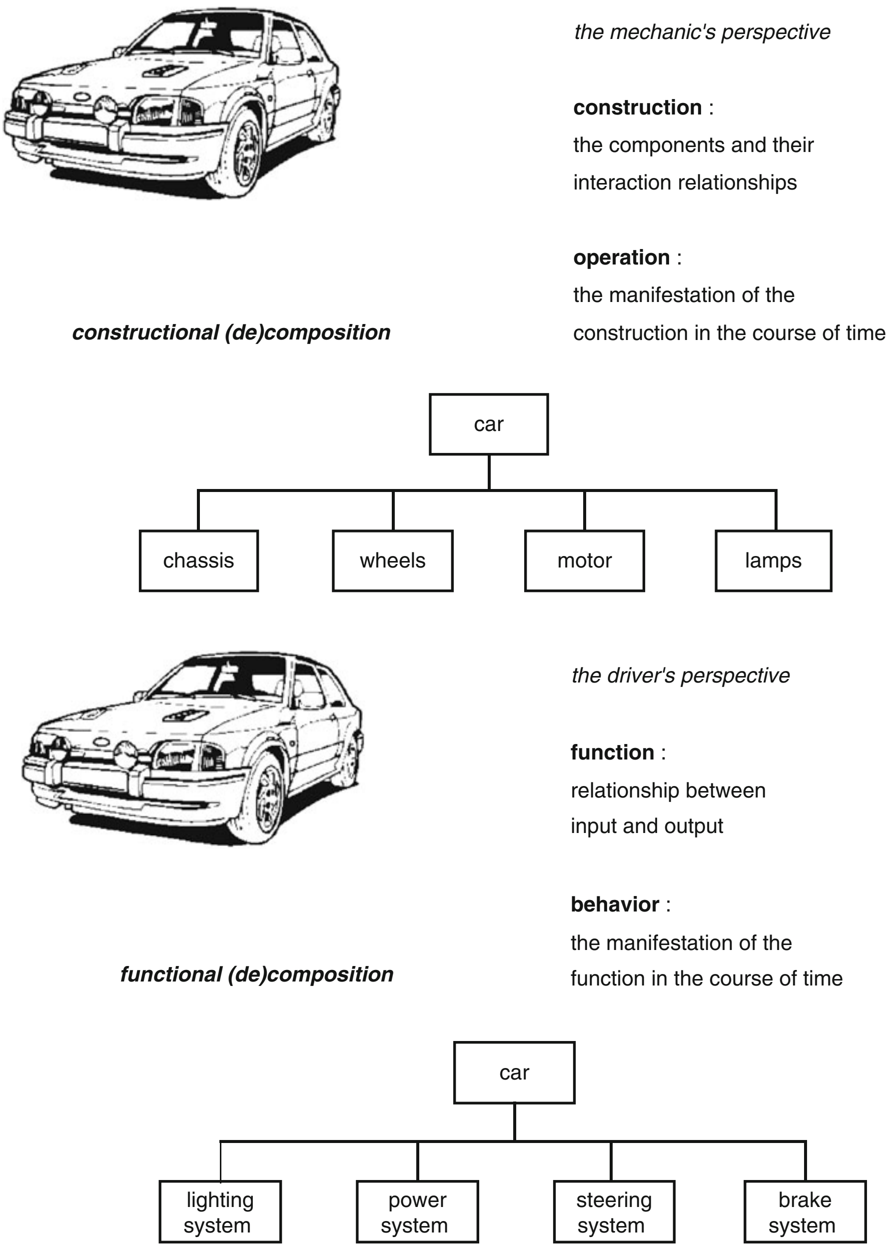


Fig. 5 White-box model and black-box model

Constructional (de)composition and *functional (de)composition* are fundamentally different. The first one applies to the car as a constructional entity and the second one applies to the car as a functional entity. There is only one constructional (de)composition, but there may be as many functional (de)compositions as there are subjects that conceive them. The *purpose* of a system is not an inherent system property but a relationship between the system and a stakeholder. The *function* of a system is a socially agreed upon purpose.

6.3 The Generic System Development Process

In engineering the *use-support* structure between systems is an important one. To say that a system S1 uses a system S2 (and conversely S2 supports S1) means that the construction of S1 uses the function of S2, which by nature is expressed in the "language" of the construction of S1. A use-support relation between two systems is neither a sub-super nor a level structure relation, but a sui generis relation.

Figure 6 exhibits the complete and generic process for developing an *object system* (OS) for the sake of supporting a *using system* (US). It consists of four phases: function design, construction design, engineering, and implementation. *Function design* starts from the construction of the US and results into the *functional model* (functional specifications) of the OS. Ideally, function design starts from the *ontology* (or ontological model) of the US. This is defined as the highest level constructional model, which is fully implementation independent. This is the only way to arrive at

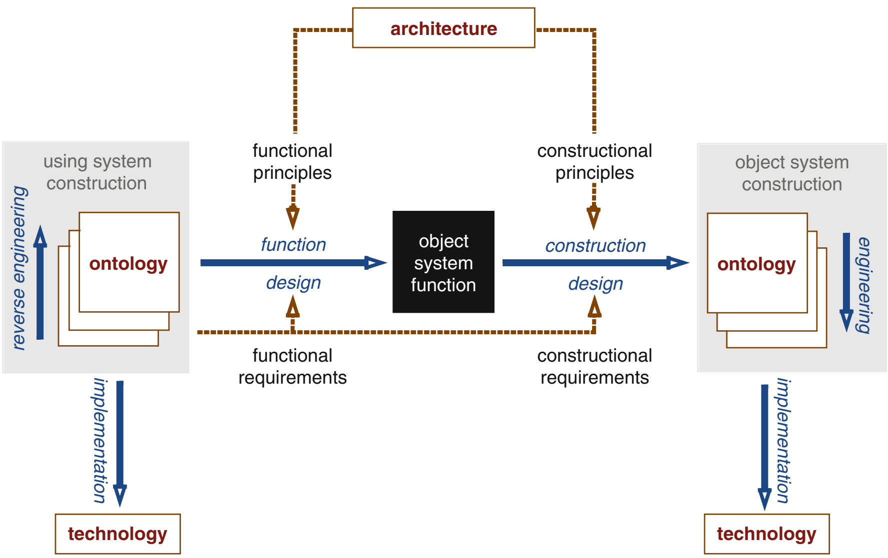


Fig. 6 The Generic System Development Process

objectively determinable (functional and constructional) requirements. If the ontological model of the US is missing, it can be produced through *reverse engineering*. Next, it is of paramount importance that the functional specifications of the OS do not contain any constructional issue.

Construction design starts from here and results into the *constructional model* (constructional specifications)of the OS. Ideally it takes place in two phases. First the *ontology* (or ontological model) of the object system is created. Ideally, construction design starts with the ontological design of the OS, followed by *engineering* (also called *technical design*). This means that one first analyzes the functional "problem" completely in implementation independent constructional terms (the ontology of the OS) before going to synthesize the "solution". Then this model is *engineered* into the *implementation model* through a series of intermediate models (model driven design). Only in this way can one objectively guarantee that an optimal *implementation model* (the lowest level constructional model) will be delivered.

Function design and construction design are generally iterative, in order to arrive at a balanced compromise between reasonable functional specifications and feasible constructional specifications.

Theoretically, *architecture* is the normative restriction of design freedom. Practically it is a coherent set of *design principles*, derived from and consistent with the strategy of the US. These principles can be divided into *functional* and *constructional* principles. They come in addition to the requirements and therefore may be conceived as the operational specification of generic requirements (next to the specific requirements regarding a particular OS).

6.4 Violation of the τ-Theory

Apparently, it is very hard for the human mind to make and maintain a sharp distinction between function and construction. For sure, one of the reasons is that in the natural languages purely constructional names are very rare. Making the distinction carefully in the process of designing is however of utmost importance. Alexander [1] provides the excellent example of the design of the one-hole kettle, which was "invented" in the 1950s of the twentieth century. Imagine that you would have the task to design a new kettle according to, among others, the next functional requirements: (1) there must me a hole to pour water in, and (2) there must be a hole to pour water out. Consider how difficult it is to come up with only one constructional element that satisfies both requirements if you are only familiar with two-hole kettles. Note also that, as a consequence, design patterns are potentially counterproductive since they deny the need for creativity in constructional design. Next, like in enterprise engineering (as we will see later), there is an urgent need in software engineering for a theory that is founded on truly elementary notions of data processing, and such that combinatorial explosions of software changes can be avoided. Normalized Systems provides such a theory [17].

In the 1970s and 1980s of the last century a lot of methodologies were developed to help manage the complex task of applying ICT to organizations, i.e. to design, engineer, and implement ICT applications. In the 1990s they were renamed into architecture frameworks, often without appreciable modifications or extensions. In order to avoid that the providers of the numerous commercial architecture frameworks become offended, we choose TOGAF, the framework of the Open Group, for closer investigation. TOGAF violates the τ-theory in many ways. First, it lacks any scientific foundation. Second, it does not distinguish clearly between function and construction, nor between functional and constructional design. Third, it does not apply the notion of system ontology, nor a sensible notion of architecture; instead it uses the term "architecture" to mean several different things. In summary, TOGAF is an incoherent and inconsistent collection of best practices. So, TOGAF is not $\phi\tau\psi$ but bullshit. Recall that the same holds for the other architecture frameworks.

7 Summary of the ψ-Theory

Next to the thorough understanding of systems, as contained in the τ-theory, a second paradigm shift was needed in the 1990s of the twentieth century, in order to bridge the gap between information systems engineering and the organizational sciences. The new paradigm is brought to us by the social sciences and by language philosophy, in particular by the LAP community.[9] It is the insight that communication is not exchanging information but primarily social interaction: while communicating, people engage into commitments.

Below, the ψ-theory is summarized[10] (the Greek letter ψ is pronounced as PSI, which stands for Performance in Social Interaction). It bridges the said gap and provides deep answers to key questions like: What is the common core of information, action and organization? How can the immense complexity of organizations be made intellectually manageable? The ψ-theory explains how and why people cooperate, and in doing so bring about the business of an enterprise. It is rooted in the information sciences, notably semiotics [21] and language philosophy [2, 19], in systemics [3], and in the social action theory [10].

7.1 Organizational Atoms

An *organization* is a social system, i.e. a system of which the elements are subjects (human beings in their capacity of social individual). They perform two kinds of acts. By performing *production acts* (or P-acts for short), the subjects contribute to bringing about products and services. A P-act can be material (like manufacturing) or immaterial (like judging). By performing *coordination acts* (or C-acts for short),

---

[9] For a good overview of the work of the LAP community, the reader is referred to the May 2006 issue of Communications of the ACM [6].

[10] For an extensive discussion of the ϕ-theory the reader is referred to [5].

subjects enter into and comply with commitments towards each other regarding the performance of P-acts. The generic structure of C-acts is exhibited in Fig. 1, which we repeat for convenience sake below. The example concerns the request by the subject John (the performer) towards the subject Mary (the addressee) to become member, e.g. of a library.

The *proposition* consists of a P-fact (in this case the start of a membership) and an associated time (in this case the start date of the membership). The *intention* represents the "social attitude" taken by the performer with respect to the proposition (in this case the request). In order to perform the exhibited C-act successfully, three conditions must be satisfied, for which John and Mary have to perform a number of communication acts. The most important condition is the *performa condition*, which means that they have to achieve social understanding. The condition is satisfied if in Mary's mind the *commitment* is evoked to respond appropriately to John's request. A prerequisite for this is satisfying the *informa condition*, which means that they have to achieve intellectual understanding. This is the case if the right conceptions of both the intention and the proposition are induced in Mary's mind. Satisfying the informa condition needs at least two communication acts: one in which John informs Mary, and the other one in which Mary confirms to have understood him. A prerequisite for this is satisfying the *forma condition*, which means that they have to achieve significational understanding. This is the case if for every communication act at the informa level the sentence uttered by John is transmitted without distortion to Mary and correctly perceived by her, and vice versa.

The notion of *actor role* serves to abstract from the subject that performs an act. It is defined as the *authority* to perform a particular kind of P-act (and the corresponding C-acts). An actor is a subject in its fulfillment of an actor role. Authority is related to *competence* and *responsibility*, as exhibited in Fig. 7. As the result of performing a C-act, a C-fact is created in the C-world. Likewise, by performing a P-act, a P-fact is created in the P-world. While performing C- and P-facts, actors take into account the current state of the C- and the P-world (represented by the dashed arrows).

The occurrence of a C-fact is a *business event* to which some actor is committed to respond. Since they include the P-act/fact it is about, C-facts are the atomic building blocks of organizations, next to the *business rules* that actors apply in dealing with business events. The notion of intention (in communication) is the key to the coherent understanding of information, action and organization: the ψ-theory provides the needed new paradigm.

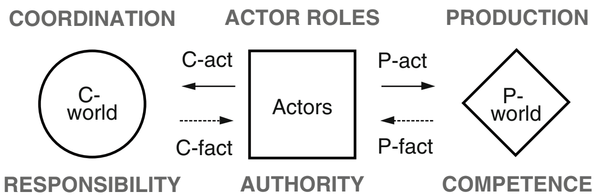


Fig. 7 Actor roles, coordination and production

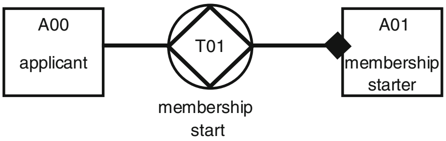


Fig. 8 The organizational building block

7.2 Organizational Molecules

C-acts and P-acts occur in a universal pattern, called *transaction*. A transaction involves two actor roles; one is the *initiator* and the other one the *executor*. The initiator starts a transaction by requesting the P-fact and ends it by accepting the delivered result. The executor promises to perform the P-act and states the achieved result. This constitutes the basic transaction pattern. The complete, universal, pattern consists of twenty different C-acts. Every transaction process is a path (possibly including iterations) through the *universal transaction pattern*. Ideally, C-acts are performed *explicitly*. They may however be performed *implicitly*, and even *tacitly* (which means that there is no observable act at all). The executor of a transaction is considered to be the *owner* of the produced P-facts.

An *elementary actor role* is the authority to be the executor of exactly one transaction kind (of which the result is one kind of P-facts, e.g. starting library memberships). Transaction kinds with their corresponding elementary actor roles constitute the molecular building blocks of organizations (see Fig. 8).

The executor (A01) of transaction kind T01 may on its turn be the initiator of one or more transaction kinds. The same reasoning holds for the executors of these transaction kinds. In this way a tree structure of transactions is constructed, called *business process*. By understanding business processes as tree structures of organizational molecules, a considerable reduction of complexity is achieved (in practice well over 70%).

7.3 Enterprise Ontology

Every organization is a layered integration of three aspect organizations, called its B-organization, its I-organization, and its D-organization, in accordance with the use-support structure (see Sect. 6). The *B-organization* comprises all transaction kinds (and their corresponding actor roles) that deliver *business services*, i.e. of which the P-act/fact is a business act/fact. These include the transactions that deliver such a service to the environment as well as the transactions that deliver components for such a service to internal actors. The *I-organization* comprises all transaction kinds (and their corresponding actor roles) that deliver informational services to the B-actors. An *informational service* concerns the remembering and the provision of B-organization P- or C-facts, including externally acquired P- or C-facts, or of derived

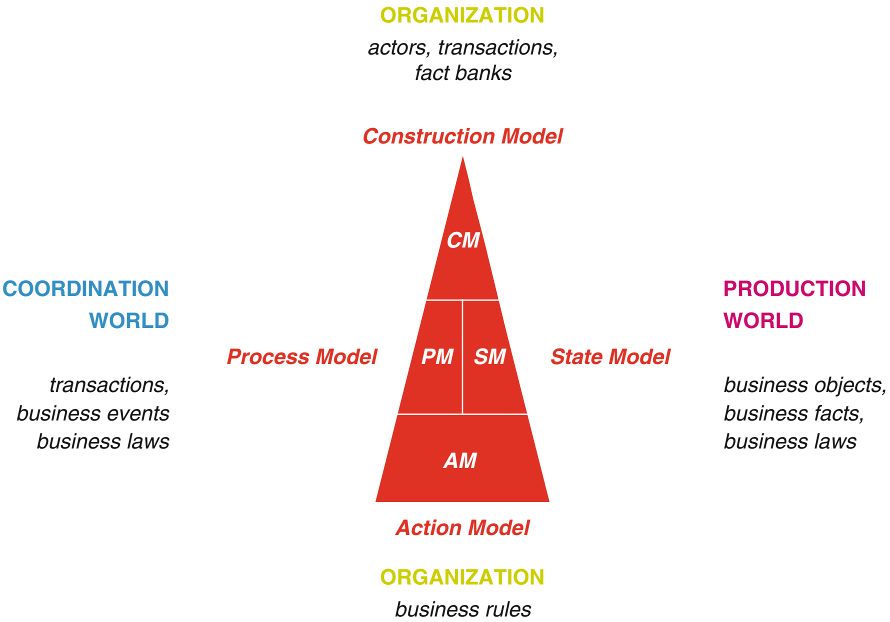


Fig. 9 The four ontological aspect models

facts based on these. The *D-organization* comprises all transaction kinds (and their corresponding actor roles) that deliver documental services to the I-actors. A *documental service* concerns the creation, the transportation, the storage, or the retrieval of a document that contains information needed by the actors in the B-organization. So, the D-organization supports the I-organization, as the I-organization supports the B-organization.

The understanding of the B-organization of an enterprise according to the ψ-theory is called the *ontology* of the enterprise. By focusing on the B-organization of an enterprise, another considerable reduction of complexity is achieved (in practice well over 70%). As is exhibited in Fig. 9, there are four distinct views on the ontology, or ontological model, of an enterprise, called the construction model (CM), the process model (PM), the state model (SM), and the action model (AM).

7.4 *Violation of the ψ-Theory*

Although the need for a coherent, consistent, comprehensive and yet concise understanding of the essence of an enterprise is recognized widely, few professionals use a methodology with such characteristics. If they would do, we would not have anymore incompatible process models and data models. We also would not have anymore useless enterprise “architectures”. Rarely does a business “architecture” contain conceptual models of the enterprise that could serve as an appropriate starting point for producing the information (systems) "architecture", as it should be the case. Lastly, we also would not call banks and insurance companies information processing enterprises, for the only reason that they have no tangible products. Immaterial B-organization P-facts are as real as material ones! At the same time, one of the key success factors of all DEMO-projects has been the ontological model of the B-organization. Particularly managers like it, because they are concise and they "speak their language". Often the construction model is sufficient for analyzing the problems at hand, and for finding satisfying solutions.

All well-known current approaches to business process modeling, like Flowchart, BPMN (Business Process Modeling Notation), EPC (Event Process Chain), and PetriNet, reduce business processes to sequences of (observable) actions and results, and human beings to human resources. No distinction is made between the B-, the I-, and the D-organization. Moreover, the resulting models lack the transaction structures, which makes them completely useless for redesign and re-engineering, if only for the fact that they are unable to detect tacitly performed coordination acts (which happen to occur quite often). So, contemporary business process modeling is not $\phi\tau\psi$ but bullshit. One better erase the predicate "business". Even worse are the function-oriented modeling techniques, like SADT and IDEF0, since by nature they reflect the personal interpretation of the modeler.

8 Conclusions

The developers/designers of contemporary (automated) information systems are not equipped to deal with the huge challenges they are facing in an adequate way. Actually they were not for a long time already, as we have seen when discussing the ϕ-theory (Sect. 5). It has been shown that the instantiation of (generic) concepts for checking their correctness is an indispensable intellectual technique: *verification by exemplification*. Let us therefore add it to Dijkstra's three techniques for appropriately performing software engineering, as mentioned in Sect. 4. By understanding and applying this technique, next to the other three, the design of information systems becomes intellectually manageable. The elementary fact, expressed in an elementary sentence, is truly the basis of conceptual modeling, as Wittgenstein already demonstrated [22].

However, even if one would understand and correctly apply the ϕ-theory, there is still the enormous gap between business and ICT. Information systems designers will still not be able to design systems that satisfy the needs of the users because of a fundamental lack of understanding. The current notion of organization is too mechanistic; business processes are reduced to sequences of actions and results. All the richness and essence that is inherent in the deep structure of business processes is ignored, as has been demonstrated while discussing the ψ-theory in Sect. 7. There is an urgent need to broaden the scope and to make the transition from information systems engineering to enterprise engineering. The Enterprise Engineering Paradigm.

As presented in Sect. 3, is a first step towards the development of this exciting new discipline.

Postulate 3 of the Enterprise Engineering Paradigm has been elaborated in the τ-theory (Sect. 6). The importance of sharply distinguishing between the function and the construction perspective is paramount. Unfortunately, it is also hard to make it; it requires designers to be very disciplined. But engineering is not an easy job; it has to be based on solid scientific grounds. This is contrary to the state-of-the-art in information systems development, which contains a lot of bullshit, as we have seen. Fighting this bullshit is a major task of any professional who takes his or her job seriously. To add another quote of Frankfurt [9]: bullshit is a greater enemy of truth than lies are.

The ontological model of an enterprise offers a total reduction of complexity of well over 90%. Moreover, it is coherent, consistent, and comprehensive. Next, it reflects the way social individuals (human beings) behave when cooperating in order to achieve some (business) goal. Therefore, validating the design of a business process or an information system on the basis of the ontological model is a prerequisite for delivering anything useful. This intellectual technique has proven to be so effective that we add it to the four techniques we have collected up to now: *validation from ontology*. By understanding and applying this technique, next to the other four, enterprise engineering becomes intellectually manageable.

References

1. Alexander C (1960) Notes on the synthesis of form. Ablex Publishing Company, Norwood
2. Austin JL (1962) How to do things with words. Harvard University Press, Cambridge
3. Bunge MA (1979) Treatise on basic philosophy, vol. 4. A world of systems. D. Reidel Publishing Company, Dordrecht
4. CACM (2006) Special issue of Communication of the ACM on the Language Action Perspective
5. Dietz JLG (2006) Enterprise ontology—theory and methodology. Springer, Heildelberg
6. Dietz JLG (2006) The deep structure of business processes. Commun ACM 49(5):59–64
7. Dietz JLG (2009) Architecture: building strategy into design. SDU Publishing, The Hague
8. Dijkstra EW (1976) A discipline of programming. Prentice-Hall, Englewood Cliffs
9. Frankfurt HG (2005) On bullshit. Princeton University Press, Princeton
10. Habermas J (1981) Theorie des Kommunikatives Handelns, Erster Band. Suhrkamp, Frankfurt am Main
11. Hoogervorst JAP, Dietz JLG (2008) Enterprise architecture in enterprise engineering. Enterp Model Inf Syst Archit 3(1):3–11, July. ISSN 1860-6059
12. Hoogervorst JAP (2009) Enterprise governance and enterprise engineering, Springer, Heildelberg. ISBN 978-3-540-92670-2
13. Kaplan RS, Norton DP (2004) Strategy maps. Harvard Business School Press, Boston
14. Kotter JP (1995) Leading change: why transformation efforts fail. Harv Bus Rev 71(2):59–67
15. Langefors B (1977) Information System Theory. Information Systems 2, 207–219
16. Lehman M (1980) Programs, life cycles, and laws and laws of software evolution. Proceedings of the IEEE 68(9), 1060–1076
17. Mannaert H, Verelst J (2009) Normalized systems—recreating information technology based on laws for software evolvability. Koppa, Belgium

18. Pettigrew A (1998) Success and failure in corporate transformation initiatives. In: Galliers RD, Baets WRJ (eds) Information technology and organizational transformation. Wiley, Chichester
19. Searle JR (1969) Speech acts, an essay in the philosophy of language. Cambridge University Press, Cambridge
20. Simons P (1987) Parts—a study in ontology. Oxford University Press, Oxford
21. Stamper RK (1973) Information in business and administrative systems. Wiley, New York
22. Wittgenstein L (1922) Tractatus logico-philosophicus (German text with an English trans: Ogden CK). Routledge, London

The Mobile and Mobility: Information, Organisations and Systems

John Traxler

Abstract Information is the basis of our society, of our businesses and of our organisations. Once, information was marginal to organisations and then gradually information became central. Consequently, information systems development methodology, "... [the] recommended collection of philosophies, phases, procedures, rules, techniques, tools, documentation, management, and training for developers of Information Systems", also became central (Avison and Fitzgerald 1988). Over the last decade, the mobility and connectedness afforded by universal personal devices, systems and technologies have meant that the production, transformation, transmission, consumption, ownership, control, nature and significance of information have changed rapidly and dramatically. The consequences for information systems, for the development of information systems, and for the organisations that use them are still unfolding. This paper outlines in very general terms the impact of mobility and connectedness and asks about the effects on information systems and their development.

1 Introduction

Personal mobile devices are curiously both pervasive and ubiquitous, both conspicuous and unobtrusive, both noteworthy and taken-for-granted in the lives of most people. Almost everyone owns one and uses one, often more than one. Not only do they own them and use them but they also invest considerable time, effort and money choosing them, buying them, customising them, enhancing them and exploiting them. These devices express part or much of their owners' values, affiliations, identity and individuality through their choice and their use. They include smart-phones, satnav, games consoles, digital cameras, media players, netbooks and handheld computers. They are a challenge to the world of information and organisation and how systems are developed.

J. Traxler (✉)
Learning Lab, University of Wolverhampton, TF2 9NT Telford, UK
e-mail: john.traxler@wlv.ac.uk

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_2,

The personal, cultural, organisational and social aspects of their impact hinge in many respects on the essential difference between desktop technologies and mobile technologies, a difference that means we can ignore the former but not the latter. Interacting with a desktop computer takes place in a bubble, in dedicated times and places where the user has their back to the rest of world for a substantial for a probably premeditated episode.

Interacting with mobile technologies is different and woven into all the times and places of users' lives. Mobile phones have created "simultaneity of place": a physical space and a virtual space of conversational interaction, and an extension of physical space, through the creation and juxtaposition of a mobile "social space", a place where conversations are no longer substantial discrete entities but instead are multiple living threads. This affects people's sense of time, space, place and location, their affiliations and loyalties to organisations and communities, the ways in which they relate to other individuals and to other organisations, their sense of their identity, and their ethics, that is their sense of what is right, what is wrong, what is approved of and what is appropriate.

Therefore, when we say we can ignore desktop technologies but not mobile technologies we mean that desktop technologies operate in their own little world, mobile technologies operate in *the* world. There are, furthermore, issues of agency, ownership and control. Desktop technologies are organizational technologies in buildings whilst mobile technologies are personal technologies with people.

We are of course evading a precise definition of both "desktop" and "mobile"; the laptop, the iPad and the netbook, for example, fall between the two extremes and the point here is not to make a hard distinction based on objective technical characteristics but a softer and more fluid distinction based on the values, perceptions and preferences of individual and organisations. The point here is also move attention to the notions of mobility and connectedness and away from those of mobiles and connectivity.

Mobile devices demolish the need to tie particular activities to particular places or particular times. They are reconfiguring the relationships between public and private spaces, and the ways in which these relationships are penetrated by mobile virtual spaces. Virtual communities and discussions had previously been mediated by static networked PCs in dedicated times, places and spaces. Now, mobile technologies propel these communities and discussions into physical public and private spaces, forcing changes and adjustments to all three as we learn to manage a more fluid environment. This is clearly a profound challenge for organisations.

The obvious starting points are the quantitative aspects of this impact. The statistics are commonplace: new phones routinely make national headlines, mp3 downloads outnumber CD sales, camera-phones outnumber cameras, smart-phones outnumber laptops, mobile phone ownership is reaching saturation and the British send over two billion texts a week.

In fact, mobility is now increasingly seen as a defining characteristic of our societies and our organisations but it is under-researched in its own right. At its broadest, we see the pervasive effects of mobility in "... five highly interdependent 'mobilities' that form and re-form diverse networks:

- Corporeal travel of people for work, leisure, family life, pleasure, migration and escape.
- Physical movement of objects delivered to producers, consumers and retailers.
- Imaginative travel elsewhere through images of places and peoples upon TV.
- Virtual travel often in real time on the internet so transcending geographical and social distance.
- Communicative travel through person-to-person messages via letters, telephone, fax and mobile" (Urry 2007, p. 47 and elsewhere).

These are all part of organisations' environment, some impact directly on their information. One implication is that location and movement can no longer be discounted or demoted as merely non-functional requirements in specifying and modelling an information system, nor can the business environment be assumed to remain as stable and coherent as it currently seems.

2 Mobiles at Work

Mobiles and mobility are changing many aspects of work, employment and the economy. The economic aspects of these trends are twofold. Firstly, the shifts in the nature of economic activity, that is in the jobs that people do, the products and services they supply, the assets and resources they invest and the businesses they work for, as mobile systems become more and more central to economies across the world. Mobile phone networks and hardware manufacturers are major multinational organisations, investing in R&D, developing products, supplying services, running call centres and employing many thousands of people, at the expense of more traditional parts of the economy, perhaps in the more developed parts of the world. Media distributors and banking operations amongst others, have adapted to the new mobile economy and trade ring-tones, downloads, airtime and credits. Secondly, the changes in the nature of work itself, in the times and places of work and the relationships within work are changing. The improved connectivity between a mobile workforce and its headquarters means greater efficiency since peripatetic workers can be deployed and supported at a distance. It also means greater supervision and increased deskilling. Furthermore since mobile technologies operate on the move as well as at a distance, we see increasing workloads as people stay connected on holidays and weekends, and we see the "day-extender syndrome" (International Telecommunications Union 2004, p. 28), weakening the home/work boundaries, as people work whilst they travel or relax.

These changes clearly affect the business processes within organisations that information systems must support, and clearly affect the possible processes by which information systems are developed.

3 Organisations in Space and Time

Mobile technologies are eroding established notions of time as a common structure. In their place we see the "approx-meeting" and the "multi-meeting" (Plant 2000, p. 31), "Our sense of time need not necessarily be strictly governed by linear time, but can instead be socially negotiated" (Sørensen et al. 2002, p. 3) and the "micro-coordination of everyday life" (Ling 2004, p. 69) alongside the "softening of schedules" (Ling 2004, p. 73) afforded by mobile devices. Nyíri (2007, p. 301) says, "with the mobile phone, time has become personalized". Agar (2003, p. 4) makes a direct comparison between the mobile phone and wrist watch, in terms of intimacy and ownership, but a direct contrast in terms of personal freedom, saying, "while it might have felt like liberation from tradition, the owner was caught anew in a more modern rationality, for, despite the fact that the pocket watch gave the owner personal access to exact time, accuracy depended on being part of a *system*", in fact it *made* the owner part of a system, handcuffed to it. Time zones and daylight saving, also artefacts of the Industrial Revolution, in aftermath of the new national railway networks, had a similar effect of creating a unified and monolithic time system but now of course personal mobile connectedness erodes that too; international travellers are no longer locked into their local time zone. Mobile phones mean they are also still tethered to family times back home and to the rhythms of their office and colleagues back at base rather than their physical location. These remarks about perceptions of time are significant given how much the notion of time figures as a major constituent of information system development methods, from structured methods through rapid methods to the current agile methods. Business process reengineering and rapid application development (RAD) are two examples of information system development with a strong and objective temporal dimension. The changes we describe not only accelerate movement along this dimension but fragment and socialise it.

Mobile devices are accelerating teleworking and also eroding physical place as a predominant attribute of space. Place is being devalued or diluted by "absent presence" (Gergen 2002), the phenomenon of physically co-located groups, in the family home or in the organisation office, all connected online elsewhere and by "simultaneity of place" (International Telecommunications Union 2004, p. 20; paraphrasing Plant 2000) created by mobile phones, a physical space and a virtual space of conversational interaction, or an extension of physical space, through the creation and juxtaposition of a mobile "social space", thereby eroding the "work-place".

Many organisations are organised around specified times and spaces, their premises and their opening hours. They are clearly less and less well aligned to the needs and behaviour of many people, their customers, clients and workers. Furthermore the increase of 24-h rolling news, off-air recording and domestic video-on-demand means that TV schedules no longer provide a synchronous and collective experience that binds informal groups of colleagues together in the way they used to a generation ago.

Mobile devices are reconfiguring the relationships between spaces, between public spaces and private ones, and the ways in which these are penetrated by mobile virtual spaces. This reconfiguration is accompanied by what goes on in those spaces. Cooper (2002, p. 22) says that the private "is no longer conceivable as what goes on, discreetly, in the life of the individual away from the public domain, or as subsequently represented in individual consciousness", "... massive changes are occurring in the nature of both public and private life and especially of the relations between them" (Sheller and Urry 2003, p. 1). "The use of these mobile sound technologies informs us about how users attempt to 'inhabit' the spaces within which they move. The use of these technologies appears to bind the disparate threads of much urban movement together, both 'filling' the spaces 'in-between' communication or meetings and structuring the spaces thus occupied" (Bull 2005, p. 344). Earlier work on the Sony Walkman came to similar conclusions, "the Walkman disturbed the boundaries between the public and private worlds" (Du Gay et al. 1997, p. 115). Organisations predicated around workers on task at their desks are no longer on solid ground, modernity has become "liquid" (Bauman 2000).

This is accompanied by a growing dislocation of time and place, in which "everything arrives without any need to depart" (Virilio 2000, p. 20). "Closer to what is far away than to what is just beside us, we are becoming progressively detached from ourselves" (Virilio 2000, p. 83). Owing to "the tendency to previsit locations, through one medium or another; to actually arrive somewhere is no longer surprising in the way that it was ... it is becoming replaced by prevision. Thus according to this logic, the mobile would be one more technique by which the world became unsurprising" (Cooper 2002, p. 26). Another personal device, the in-car sat-nav, has a similar effect, that of *previsiting* places and locations. Other personal digital devices, the camcorder, the camera, allow us to recreate the past, to *revisit* places and locations. Google Maps with Street View also dilutes the here-and-now, creating perhaps *absent presents*, but the ever-growing sense of surveillance is implicit too.

4 Mobiles and Individual Identity in Organisations

In 2007, Charlie Schlick, Product Manager of Nokia, described company practice in talking of mobile phones as "our new private parts". These devices are personal, universal and closely linked to identity and in talking about mobile devices; we must recognize how closely they are bound up with a changing sense of self. Some authors describe personal mobile devices as becoming prosthetic; Pertierra (2005, p. 27) says, "Unlike desktops and other immobile technologies, mobile phones more closely resemble tools or prosthetic devices as extensions of the body. They become extensions of the hand, allowing us to connect anytime, anywhere, with anybody. Bodies themselves become writing devices as phoneurs negotiate new urban spaces". Other authors describe them as becoming "embodied" (for example, Rettie 2005).

> From pacifier, to nipple, to digital umbilical cord, the mobile phone rapidly progressed to assume a vital place in the virtual biology of urban information societies of the late twentieth century. At the final extreme, the mobile phone's connectivity might be completely subsumed into the body, and all other forms of communication become redundant email, web, phone calls, all can be delivered over the universal handheld (Townsend 2001, p. 70).

"One can be interrupted or interrupt friends and colleagues at any time. Individuals live in the phonespace—they can never let it go, because it is their primary link to the temporally, spatially fragmented network of friends and colleagues they have constructed for themselves. It has become their new umbilical cord, pulling the information society's digital infrastructure into their very bodies. In fact, as technical evangelists at Nokia pondered, mobile communications could eventually evolve into an activity indistinguishable from telepathy" (Townsend 2001, p. 70). However there is also a widespread perception that technology, including mobiles, is enabling supervision, oversight and surveillance and this must erode or weaken aspects of identity, or perhaps merge and confuse professional identity and social identity, digital identity and physical identity.

Mobile devices have been associated with new forms of discourse and thus with different communities. The obvious example is "text-speak" and its original subversive association with teenagers but another is the evolution of the "missed call" around the world (Donner 2008). There are also more subtle transformations. Goffman (1971), for example, noted the phenomenon of "civil inattention", where in certain social situations it is customary and necessary not only to not speak to others but to avoid looking directly at them. This management of gaze is one way in which the boundary between public and private is negotiated and is now often a characteristic of creating a private space for mobile phone conversations in a public setting; a similar concept is the "tie-sign", the various gestures that keep a face-to-face encounter live and "in play" whilst servicing an interruption caused by a mobile phone call. The recipient of the call is obliged to "play out collusive gestures of impatience, derogation, and exasperation" according to Goffman.

Murtagh (2002) describes a wide set of non-verbal actions and interactions with the mobile phone in public, and these are part of a wider transformation of discourse and social interaction as society engages with mobile technologies. Organisations are growing less likely to know what is going on.

Mobile devices also affect many aspects of the processes by which knowledge, ideas, images and information are produced, stored, distributed, delivered and consumed. They are now part of a system that allows everyone to generate and transmit content not just passively store and consume it, making mobile systems an integral part of the Web2.0 ideology that takes users from being merely the Web's readers to its writers. This happens in several ways, for example is citizen-journalism, the phenomenon of people using their camera-phones to capture news events and then using perhaps YouTube or Flickr to broadcast the images and comments, with no intervention or control from head office, the centralized government, media or news corporations. Organisations will have less control about what individuals show or say about them (though will quickly explore ways to appropriate and colonise these new media).

A more general example includes Flickr, YouTube, Wikipedia and other file-sharing or wiki-based technologies that have migrated onto mobile devices as connectivity and usability improve. Once built into mobile devices, these technologies are starting to exploit the capacity to capture or retrieve information that is context-aware and location-specific. Google on mobiles, for example, offers an improved "local search experience" based on the expectation that there is a market for area information such as cinema listings. Content, information and knowledge will become location-specific and this could lead to much richer, more diverse and more economically viable forms of context-aware knowledge and information.

The significance of social network technologies, such as LinkedIn or Plaxo, in facilitating virtual communities has been widely documented (Bryant 2006) and is already being exploited or appropriated by organisations. These technologies have now migrated from desktop computers to mobile devices and supplement technologies that are "native" to mobile devices, systems such as Twitter, micro-blogging systems that connect communities on the move. Multi-user virtual worlds such as Second Life will take on a mobile dimension soon. These changes will further interweave physical and virtual communities and spaces, and identities. They facilitate the creation and support of discursive communities, within and across organisations, able to collaborate whilst moving (again linking to the "smart mobs" concept (Rheingold 2002). The use of Second Life in information systems development has already begun.

One challenge for information systems development is the growing chasm between the solidity and boundedness of organisations such as companies, institutions and governments and the fluidity and movement of people's information lives, echoed in the open source development community and cloud computing.

Mobile devices will consequently soon break down the notion of stable and commonly accepted and understood corpus of knowledge and information distributed through privileged channels within or between organisations by sanctioned individuals—now everyone can produce information content, and everyone one can discuss it *anywhere/anytime* and *just-in-time*, *just-for-them*.

5 Information as Content

Mobile devices deliver knowledge "chunked", structured and connected in very different ways from the presentation, the web and the manual. Knowledge is not purely abstract, unaffected by how it is stored, transmitted or consumed. In its earliest forms, knowledge and information came from the talk, idealised as a substantial linear format from an authority with no facility to pause or rewind, and from the book, also authoritative, substantial and linear but segmented and randomly-accessed. The delivery of knowledge and information by networked computers using the Web meant a break from linearity with the introduction of hyperlinks and the need for new heuristics of usability (for example, Nielsen 1992) that described how knowledge and information should be best "chunked" and presented. We have to recognise of course the tension between a rational objective articulation of these heuristics and the capacity of new technologies to overturn them, in for example the case of the badly designed but wildly popular iPad (Budiu and Nielsen 2010).

With mobile technologies, using a small screen and a limited input medium, the "chunks" become much smaller but the navigational overhead has become much,

much larger. In essence, small pieces of information can be easily presented but their relationship to each other and to anything else may be difficult to understand, thereby fragmenting and perhaps trivialising what people know. As Marshall McLuhan (McLuhan and Zingrone 1997) says, “It is the framework which changes with each new technology and not just the picture within the frame”.

Mobile technologies are starting to merge with an equally powerful technology, that of “cloud” computing (as described in Wiess 2007). This is the phenomenon of data, applications and processing moving away from specific hardware hosts and into the Internet. Google docs and Flickr are examples. The combined consequence for organisations and institutions will be to challenge the primacy of organisationally controlled desktop computers. A different medium-term trend will be for these activities to move into the environment, into buildings, furniture, vehicles or clothing, and to become ambient and pervasive (Satyanarayanan 2001). The consequence for organisations will be to accelerate the convergence of physical architecture and virtual architecture, and to blur the boundaries between institutional space, social space and personal space. This has implications for information systems development methods that focus on the immediate environment for developer teams (Martin 1991).

In another respect, information and knowledge are no longer anchored in physical artefacts. The advent of eBook readers and mp3 media players, for example, means that books and records are longer necessary to store and transmit literature and music. Video-on-demand is another part of the transformation of live social performance into consumable artefact and now into disembodied asset.

Whilst mobile technologies, especially as portals into cloud computing, seem to increase the participation and ownership of information, they may be transferring these from the jurisdictions of national governments and located institutions to powerful supra-national organisations rather than creating more democratic forms. They may also be creating more local, partial and transient forms of information as well as, obviously, catalysing exponential growth in its sheer volume.

These are all components of what has been called an “epistemological revolution” (for example, in the sense broadly outlined in Des Bordes and Ferdi 2008), a phrase used to express the fact that computers and now mobile technologies are revolutionising what we know and how we know it. In talking in these terms, we should however be careful not to obscure the nuances and differences between individual devices and technologies and the various ways in which different cultures and organisations with society adopt and adapt them.

6 Conclusion

This account is only partial and attempts to describe a fragmented and rapidly changing picture. The organisational world, its information systems and their development are very complex. These make sweeping statements or recommendations fairly problematic. The literature described here does however systematise and legitimise what we can easily see around us in our colleagues and organisations, and creates a broader context for the future of information systems development.

References

Agar J (2003) Constant touch. Icon, Cambridge
Avison DE, Fitzgerald G (Paperback, 1988) Information systems development: Methodologies, techniques and tools. McGraw-Hill
Bauman Z (2000) Liquid modernity. Polity, Cambridge
Bryant T (2006) Social software in academia. Educause Q 29(2):61–64
Budiu R, Nielsen J (2010) Usability of iPad apps and websites. In: First research findings. Nielsen Norman Group, Fremont. http://www.nngroup.com/reports/mobile/ipad/
Bull M (2005) No dead air! The iPod and the culture of mobile listening. Leisure Stud 24(4):343–356
Buscher M, Urry J (2009) Mobile methods and the empirical. Eur J Soc Theory 12(1):99–116
Caron A, Caronia L (2007) Moving cultures—mobile communication in everyday life. McGill-Queen's University Press, Montreal
Cooper G (2002) The mutable world: social theory. In: Brown B, Green N, Harper R (eds) Wireless world: social and interactional aspects of the mobile world. Springer, London
Des Bordes A, Ferdi S (2008) Do knowledge and new technologies need a new epistemology? In: Proceedings of BOBCATSSS 2008 providing access to information for everyone, Zadar, Croatia. http://edoc.huberlin.de/conferences/bobcatsss2008/
Donner J (2008) The rules of beeping: exchanging messages via intentional "missed calls" on mobile phones. JCMC 13(2008):1–22
Du Gay P, Hall S, Janes L, Nequs K (1997) Doing cultural studies: the story of the Sony Walkman. Sage, London
Gant D, Kiesler S (2002) Blurring the boundaries: cell phones, mobility, and the line between work and personal life. In: Brown B, Green N, Harper R (eds) Wireless world—social and interactional aspects of the mobile world. Springer, London, pp 121–131
Gergen KJ (2002) The challenge of absent presence. In: Katz JE, Aakhus MA (eds) Perpetual contact. Mobile communication, private talk, public performance. Cambridge University Press, Cambridge, pp 227–241
Geser H (2004) Towards a sociological theory of the mobile phone. University of Zurich, Zurich
Goffman E (1971), Relations in public. Allen Lane, Harmondsworth
Goggin G (2006) Cell phone culture. Routledge, Abingdon
International Telecommunications Union (2004) Social and human considerations for a more mobile world. In: Report from ITU/MIC Workshop on Shaping the Future Mobile Information Society, Seoul, 26 Feb 2004
Katz JE (2006) Magic in the air—mobile communication and the transformation of social life. Transaction, New Brunswick
Katz JE, Aakhus M (eds) (2002) Perpetual contact—mobile communications, private talk, public performance. Cambridge University Press, Cambridge
Lindlof T (ed) (1996) Constructing Sage, Beverly Hills, pp 127–140
Ling R (2004) The mobile connection—the cell phone's impact on society. Kaufmann, San Francisco
Ling R (2008) New tech, new ties—how mobile communication is reshaping social cohesion. MIT Press, Cambridge
Ling R, Donner J (2009) Mobile communication. Polity, Cambridge
Martin J (1991) Rapid application development. Macmillan, Indianapolis
May H, Hearn G (2005) The mobile phone as media. International J Cult Stud 8(2):195–211
McLuhan E, Zingrone F (1997) Essential McLuhan. Routledge, London

Murtagh G (2002) Seeing the "rules": preliminary observations of action, interaction and mobile phone use. In: Brown B, Green N, Harper R (eds) Wireless world: social and interactional aspects of the mobile world. Springer, London
Nielsen J (1992) Finding usability problems through heuristic evaluation. In: Proceedings of the SIGCHI conference on Human Factors in computing systems, ACM, Monterey, pp 373–380. http://doi.acm.org/10.1145/142750.142834
Nyíri K (2007) Mobile studies: paradigms and perspectives. Passagen, Vienna
Pertierra R (2005) Mobile phones, identity and discursive intimacy. Human Technology 1(1):23–44
Plant S (2001) On the mobile—the effects of mobile telephones on individual and social life. Motorola, Schaumburg
Rettie R (2005) Presence and Embodiment in Mobile Phone Communication. J Psychol 3(1):16–34
Rheingold H (2002) Smart mobs—the next social revolution. Persius, Cambridge
Rogers E (2003) Diffusion of innovations. Free Press, New York
Satyanarayanan M (2001) Pervasive computing: vision and challenges. Pers Commun IEEE 8(4):10–17
Sheller M (2004) Mobile publics: beyond the network perspective. Environ Planning D Soc Space 22:39–52
Sheller M, Urry J (2003) Mobile transformations of 'public' and 'private' life. Theor Cult Soc 20:107–125
Sherry J, Salvador T (2002) Running and grimacing: the struggle for balance in mobile work. In: Brown B, Green N, Harper R (eds) Wireless world—social and interactional aspects of the mobile world. Springer, London, pp 108–120
Sky (2009) http://news.sky.com/skynews/Home/UK-News/Microsoft-Survey-Shows-Employees-Using-Work-Time- To-Socialise-And-Work-When-They-Get-Home/Article/2009102154030-59?f=rss. Accessed 10 Oct 2009
Sørensen C, Mathiassen L, Kakihara M (2002) Mobile services: functional diversity and overload. In: Presented at new perspectives on 21st-century communications, Budapest, Hungary, 24–25 May 2002
Stald G (2008) Mobile identity: youth, identity, and mobile communication media. In: Buckingham D (ed) Youth, identity, and digital media. The John D. and Catherine T. MacArthur Foundation Series on Digital Media and Learning. MIT Press, Cambridge, pp 143–164
Townsend AM (2001) Mobile communications in the twenty-first century city. In Brown B, Green N, Harper R (eds) Wireless world—social and interactional aspects of the mobile world. Springer, London
Urry J (2007) Mobilities. Polity, London
Virilio P (2000) Polar inertia. Sage, London
Wiess A (2007) Cloud computing: PC functions move onto the web. netWorker 11(4):16–25 (December)

Design of Situational Artefacts—Conceptual Foundations and Their Application to IT/Business Alignment

Robert Winter

Abstract IT/business alignment is consistently rated as the #1 problem for CIOs of all kinds and sizes of companies. Although IS research has proposed a plethora of hypotheses and artefacts to explain and approach IT/business alignment problems, the large variety of goals and context factors seem to prevent effective solutions: Since companies face diverse challenges in achieving a high degree of IT/business alignment, a universal "one size fits all" approach seems not to be appropriate. IT/business alignment is therefore regarded as a class of relevant, important design problems that requires a situational solution approach. The conceptual foundations of situational problem analysis and artefact design are summarized, and a generic approach is proposed that transforms problem analysis into artefact design. The approach is then applied by decomposing the IT/business alignment problem into tangible qualities for business, IT systems, and IT governance. An explorative survey among 174 professionals is used to identify four clusters of IT/business alignment problems which each represent the current state according to certain qualities and also the priorities for future development. Using the proposed situational solution approach, four reference solutions are outlined.

1 Introduction

IT/business alignment has constantly been among the top priorities for IT executives (Luftman 2005; Luftman et al. 2006, 2009; Luftman and Kempaiah 2008). On the one hand, the IT/business alignment phenomenon needs to be better understood. On the other hand, artefacts need to be constructed on the foundation of that understanding which enhance or extend IT/business alignment. We concentrate on the latter research perspective. Our analysis of related work, i.e. existing methods and models

R. Winter (✉)
Institute of Information Management, University of St. Gallen, Müller-Friedberg-Strasse 8, 9000 St. Gallen, Switzerland
e-mail: Robert.Winter@unisg.ch

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_3,

for IT/business alignment, in Sect. 2 shows that different problem contexts (e.g. industry, company size) and different design objectives (e.g. improving transparency, reducing inconsistencies, improving flexibility) are usually not considered. As diverse problem situations require different solution designs, design research should provide situated or at least adaptable solutions instead of “one-size-fits-all” artefacts and might thereby enhance or extend their utility. As a precondition for constructing such solutions, however, precise specifications of relevant IT/business alignment situations are needed. From that knowledge about problem situations, important aspects of proposed solutions can then be derived.

The contribution of this chapter is to propose a transformation between situational problem analysis and situational artefact construction. Since IT/business alignment is a diverse and widely unsolved problem, this class of design problems is used to instantiate the proposed approach, i.e. to identify problem situations and derive situational solution artefacts for IT/business alignment.

The remainder of this chapter is structured as follows: Sect. 2 presents and discusses related work on IT/business alignment. The often “fuzzy” understanding of IT/business alignment is operationalized by considering qualities of IT systems, business, and IT governance. In Sect. 3, we present conceptual foundations of situational problem analysis and situational artefact design. Based on a broad empirical analysis, four IT/business alignment situations are presented in Sect. 4. The transformation of situational problem analysis into appropriate solution artefacts for IT/business alignment is outlined in Sect. 5. The findings are discussed, and an outlook on future work is given in Sect. 6.

2 Related Work on IT/Business Alignment

The framework proposed by Henderson and Venkatraman (1999) is widely applied and is generally considered to be a key reference alignment model. It contains four components needed for alignment assessment, viz. (1) business strategy, (2) IT strategy, (3) organizational infrastructure and processes, and (4) IT infrastructure and processes. The interactions of these four components along two dimensions, strategic integration and functional integration, form the state space of the model. Neither do the authors propose problem solution artefacts, nor do they consider situational differences of IT/business alignment problems.

Luftman (2000) develops a strategic alignment measurement tool based on detailed maturity models covering six distinct areas: (1) communications, (2) competence/value, (3) governance, (4) partnership, (5) technology scope, and (6) skills maturity. This work was later extended in Sledgianowski and Luftman (2005) into the Strategic Alignment Maturity Assessment (SAMA). While an assessment tool can be considered as an artefact that supports problem solution, Luftman’s approach has no situational characteristics.

Aier and Winter (2009) propose what they call *virtual decoupling* to achieve IT/business alignment, i.e. an architecture-centric approach that separates the external view on architecture from its particular implementation. In doing so, the level of complexity is significantly reduced, thus enabling a more rational and less myopic approach to managing the IT/business alignment. Aier and Winter do however not consider situational differences of IT/business alignment problems.

Plazaola et al. (2007) attempt to address strategic IT/business alignment by enterprise architecture management. They propose a meta model and explain the criteria as well as the process for connecting Luftman's strategic alignment measures (Luftman 2000) with Zachman's Enterprise Architectural Framework (Zachman 1987). However, the proposed meta model does not include any guidelines for how to apply it in particular enterprises and adjust it to the specific problem setup.

Based on an empirical study, Baumöl (2006) argues that IT/business alignment can only be achieved by using a systematic change method. She proposes such a method, called Change Method Engineering, which is based on empirical findings of critical IT/business alignment success factors. The approach is based on a taxonomy of transformation problems and thus can be considered as situational. The situations are however defined regarding general transformation and not specifically regarding IT/business alignment.

Although Baumöl (2006) at least attempts to provide a situational solution to IT/business alignment, all related work fails in addressing specific situational characteristics of IT/business alignment by appropriate solution artefacts. Moreover, related work exhibits a wide array of more or less fuzzy definitions of the IT/business alignment phenomenon. In order to specify IT/business alignment in a way that allows for systematic analysis and solution construction, we refer to an approach developed at KTH Stockholm. While Johnson and Ekstedt (2007) propose a set of measurable IT system qualities, Gammelgård et al. (2006) propose a set of measurable business and IT governance qualities that is aggregated later e.g. by Gustafsson et al. (2008). Figure 1 illustrates the conceptual analysis framework for IT/business alignment. In particular, the jigsaw puzzle metaphor is intended to convey that no single set of qualities has primacy over any other. IT/business alignment arises in the interaction of all the qualities and can be achieved only by adjusting all of them so that they fit each other. A comprehensive documentation of the qualities can be found e.g. in Saat et al. (2011).

3 Situational Problem Analysis and Situational Artefact Design

Design Science Research is a research paradigm that has been, among other application domains, successfully deployed to Information Systems (IS). In the following, we will use DSR to abbreviate Design Science Research in Information Systems. At its core, DSR is about the rigorous construction of useful IS artefacts, i.e. constructs, models, methods, or instantiations (March and Smith 1995). Hevner et al.

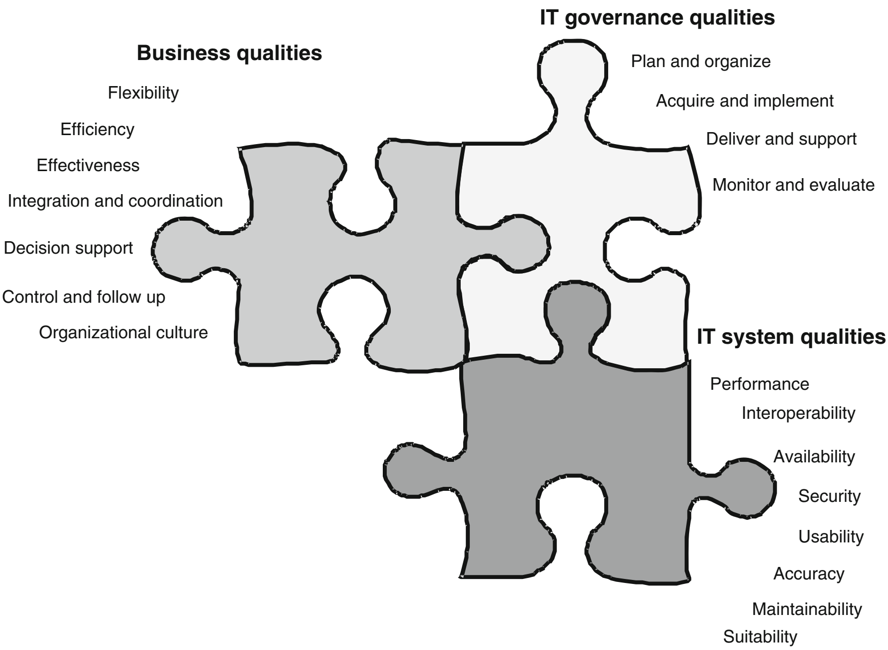


Fig. 1 A conceptual view of an IT/business alignment operationalization. (Saat et al. 2011)

(2004) generalize constructs, models, methods and instantiations as "technology-based solutions to important and relevant business problems". Design problems in organisations are generically defined as "the differences between a goal state and the current state of a system" (Hevner et al. 2004).

Besides of being important and relevant, the design problem—and hence the proposed design solution (= DSR output artefact)—should be sufficiently general. For Hevner et al. (2004), generality is one of three quality criteria of an DSR artefact. Baskerville et al. (2009) demand a design research artefact to "represent [.] a general solution to a class of problems". In the following, we will therefore assume that DSR results are generic (and not specific) IS artefacts which are useful for solving a class of design problems.

The two research goals of generality and utility should be differentiated. In their research on reference modelling, Becker et al. discuss the *reference modelling dilemma*: "On the one hand, customers will choose a reference model that [. . .] provides the best fit to their individual requirements and therefore implies the least need for changes. On the other hand, a restriction of the generality of the model results in higher turn-over risks because of smaller sales markets" (Becker et al. 2007). This dilemma is not only apparent in reference modelling, but also exists for other general solutions to classes of design problems (e.g. methods). In a very simple form, it can be formalized as

$$U^* g = c$$

where U denotes an artefact's utility from a single organisation's perspective, g denotes its generality, and c is a constant (Winter 2011). With increasing generality, the individual utility of a solution for solving a specific design problem decreases—and vice versa. The sum of individual utilities is however increasing when solutions are specified on a higher level of genericity—but individual organizations might not be interested in this "general utility".

As a solution to the reference modelling dilemma, Becker et al. (2002, 2007) propose adaptation mechanisms that instantiate a generic reference model according to the specific design problem at hand. We hold a larger view, i.e. refer to all four artefact types identified by March and Smith (1995) and hence designate the extension of generic artefacts by adaptation mechanisms as *situational artefact construction* (SAC). In addition to situational reference modelling (e.g. Becker et al. 2007), situational artefact construction has also been investigated for methods (situational method engineering, see e.g. Bucher et al. 2007).

As SAC allows the researcher to develop artefacts which are adaptable to different design problems within a problem class, a crucial decision during the construction phase is to delineate the range of addressed design problems (i.e. to specify the design problem class) and to understand the relevant design situations within this class. If a design problem class is understood as a set of "similar" design problems, a design situation can be understood as a subset of design problems which are even more similar. Depending on the degree of generality, a design problem class can be partitioned into few, very generic design situations or a larger number of (different) design situations of lesser generality.

Winter (2011) extended the initial procedure proposal of Bucher and Klesse (2006) for situational problem analysis by differentiating more components and assuming that, in general, only adaptable artefacts on a certain level of generality are constructed:

1. A rough idea about the delineation of the design problem class is developed. Results of this step are definitions, a description of the system under analysis and an idea about design goals for this class of design problems.
2. A literature analysis is conducted in order to identify potential contingency factors for that class of design problems.
3. A field study is conducted in order to analyze design problems of that class in practice. As a result, the list of potential contingency factor candidates is reduced to a smaller set of relevant "design factors". Design factors might be aggregates of several contingency factors that need to be semantically interpreted.
4. The design problem class is redefined by specifying value ranges for the design factors. This means that "outliers" are ignored from further analysis in order to create homogeneous problem sets and subsets.
5. Those field study data of observations which still belong to the redefined design problem class, are used to calculate ultrametric distances between specific design problems. The calculation is based on certain "similarity" metrics—usually the Euclidian distance with regard to the observed values of design factors.

6. A useful level of solution generality is determined. Usually clustering errors related to the number of clusters are used for this analysis.
7. Using the desired solution generality, the resulting design situations are specified. The situations should not only be specified formally (by value ranges of the design factors), but also should be interpreted semantically ("design problem types").

In Winter (2011), the discovery of four design problem types for data warehousing service provision is used as an illustrative example for steps (1) through (7).

Following this approach, the design problem class can be sufficiently analyzed to allow for the systematic development of solution artefacts. Figure 2 depicts an idealized graphical visualization of ultrametric distances between design problems (C1...C33) within a design problem class C. An ultrametric distance graph supports an intuitive understanding of how closely related (or how different) design problems in a class are Winter (2011). For example, the ultrametric distance between C11 and C15 is much smaller than that between C11 and C6 (i.e., C11 is more similar to C15 than to C6) because the common solution to C11...C15 is of lesser generality than a common solution to C1 . . . C15.

It also becomes intuitively clear that solution artefacts can be constructed on different levels of generality—the fewer artefacts are to be constructed, the higher their generality has to be. For the exemplary design problem class whose ultrametric distances are illustrated by Fig. 2, nearly any number of solution artefacts between 1 (one "one size fits all" generic solution for *C*) and 33 (one specific solution for every single design problem *Cxx*) could be constructed.

For the exemplary design problem class illustrated in Fig. 2, we assume that the analysis yields a desired generality level of G_c. For G_c, the ultrametric distance graph

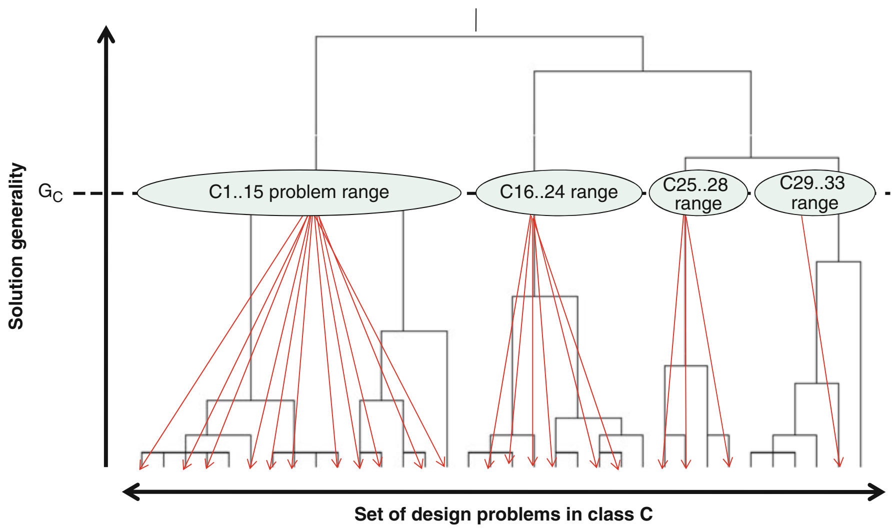


Fig. 2 Problem class decomposition into design situations. (Winter 2011)

yields four design situations (one for problems *C1...15*, one for problems *C16...24*, one for problems *C25...28* and one for problems *C29...33*). Instead of developing only one "one size fits all" solution for problem class *C*, four solution artefacts would need to be constructed that cover the respective design situation. In order to provide a solution to one of the concrete design problems within *C*, the respective situational solution artefact would have to be adapted (directed arcs in Fig. 2). The graphical representation shows that such an adaption would be much more targeted and much less costly than having to adapt a completely generic solution artefact for problems of type *C*.

The technique presented in Winter (2011) groups design problems into design situations that are homogeneous with regard to certain *design factors*, while separating design problems that are heterogeneous regarding such design factors. The design factors are not part of the initial design problem or problem class definition, but are instead the result of a principal component analysis on the problem description data. For typical design problem classes, a number of four to eight design factors are identified which explain the variance of the problem descriptions sufficiently. Every problem description can be interpreted as a point in an n-dimensional space which is implied by the n discovered design factors. A cluster analysis is then applied to identify m design situations which group these points into clusters of high intra-cluster homogeneity and inter-cluster heterogeneity.

An important DSR step is to semantically interpret the identified n design factors (= principal components of the problem data set) and m design situations (= clusters in the n-dimensional design factor space). For that purpose, it is often helpful to find out which subset of design factors characterizes a cluster best, i.e. which factor values are particularly high or low in a cluster and have a small deviation.

Understanding the problem-oriented meaning of design factors and design situations is essential for the construction of respective solution artefacts: Elementary problem solution fragments can be interpreted as "movements" in the 1-dimensional or 2-dimensional space implied by 1 or 2 relevant design factors, and complete problem solutions can be interpreted as p-dimensional vectors in the space implied by p relevant design factors for the respective solution ($p \leq n$).

Summing up, the following procedure is proposed for situational solution design (and complements the situational problem analysis procedure):

8. By linking back analysis results to the characteristics of the underlying design problem descriptions, design factors (= result of principal component analysis) and design situations (= result of agglomerative cluster analysis) of the design problem class are interpreted semantically. This step is basically a more detailed and solution driven analysis than step 7.
9. Those combinations of design (problem) situations and 1 or 2 design factors are identified where the selected design factors represent best an elementary problem solution fragment. For example, if "widespread usage of concept X within the organization" is identified as a design factor and two design situations differ significantly with regard to this design factor (cluster A = low values, cluster

B = high values), then "extend use of concept X" would be a problem solution fragment that effects a "movement" from cluster A to cluster B.

10. By linking desired properties to design factors, ideal problem solutions are related to the n-dimensional system. If many observations in the data set have already reached such desired state(s), there might be even one (rarely more) "design situation(s)" where the problem has already been solved—i.e. there might have been cluster(s) identified that represent(s) ideal problem solutions. Normally, there will be no such "problem already solved" clusters.
11. For every design situation (except "problem already solved" clusters) and every ideal problem solution, all applicable solution fragments from (9) are aggregated into a solution path. Semantically, the activities which are represented by elementary "movements" in (9) are aggregated into an activity set. This activity set is usually not only related to 1 or 2 design factors, but involves many or even all identified design factors. If the there are too many design situations and ideal problem solutions to enumerate this step, the most relevant "paths" from as-is to to-be can be extracted from survey data as long as to-be specifications—and not only as-is specifications—are including in the analysis.

4 Problem Analysis: Four Types of IT/Business Alignment Problems

This chapter summarizes Saat et al. (2011) who present findings of an online survey whose aim was, among others, to identify design situations for IT/business alignment. The survey was distributed among professionals in Sweden, Switzerland, Austria and Germany. Out of 1,105 invitations sent, 92 emails bounced, 339 persons started and 174 persons completed the survey. The online survey was active for ten days (Sept. 11–21, 2009). The survey included a final question regarding the respondent's confidence with his or her answers. Twelve persons stated weak confidence so their answers were not further considered. A total of 162 completely filled in surveys are subjected to the following analysis. Figure 3 describes the dataset in detail.

The survey is comprised of four parts: Part one contains questions regarding the background of the respondents such as industry, country, and company size as well as the respondent's role and experience. The second part of the survey has two sections. The Sect. 1 contains questions addressing enterprise architecture use for IT/business alignment and the importance and perceived maturity of IT/business alignment at the respondent's company. Section 2 contains more detailed questions related to IT/business alignment and the positioning of the IT department within the respondent's company. The third part of the survey addresses the qualities regarding IT, business, and IT governance as presented in Sect. 2. For each quality one statement was posted and the respondents were asked to mark the actual (as-is) situation (degree of realization) and desired (to-be) situation (importance for future realization) on a five-point Likert scale (where 1 equals very low, 2 equals low, 3 equals medium, 4 equals high, and 5 equals very high). The final part of the survey contains a

Industry	Count	Percent
Financial Industry	49	30.25
Utility	23	14.20
Manufacturing	17	10.49
Other	13	8.02
Transportation	12	7.41
Telecommunication	10	6.17
Software industry	8	4.94
Public sector	7	4.32
Defense/Military	7	4.32
IT/Management-Consulting	4	2.47
Healthcare/Pharmaceutical	3	1.85
Engineering/Architecture	3	1.85
Academia/Research	2	1.23
Tourism	2	1.23
Retail	2	1.23

Size of Comany	Count	Percent
<100	15	9.26
100-1.000	32	19.75
1.001-5.000	33	20.37
5.001-10.000	15	9.26
10.001-25.000	28	17.28
>25.000	39	24.07

Area of Activity	Count	Percent
Business Management	28	17.28
Business Unit	22	13.58
IT Department	52	32.10
IT Management	60	37.04

Country	Count	Percent
Austria	4	2.47
Finland	1	0.62
Germany	28	17.28
Sweden	73	45.06
Switzerland	56	34.57

n = 162

Fig. 3 Data set description. (Saat et al. 2011)

question regarding the respondents' confidence regarding the answers as well as the possibility to submit questions and feedback to the authors.

According to the proposed procedure model, the data is first examined by exploratory factor analysis (Thompson 2004) in order to identify design factors with a large explanatory power. Second, these factors are used as input for a hierarchical agglomerative clustering procedure (Härdle and Simar 2003) that groups the problem descriptions represented by the respondent's data into a small number of design situations with different characteristics.

The sampling adequacy is evaluated using the Kaiser-Meyer-Olkin (KMO) measure. In the case presented, the KMO is 0.918, which Kaiser and Rice (1974) regard to be "marvelous". The Kaiser-Guttmann criterion (Kaiser and Dickman 1959) and the elbow criterion (Cattell 1966) is used to obtain a desired number of factors. According to Kaiser and Dickman (1959) the number of factors to be extracted should be equal to the number of factors with eigenvalues bigger than one. The elbow criterion can be satisfied using a scree test as proposed by Cattell (1966). For the data set at hand, the optimal number of design factors is computed to be three.

Principal component analysis is used to extract the factors. The goal of this approach is to identify few but independent factors that reduce the items according to common characteristics. Table 1 depicts the result of the factor analysis. In order to better understand and interpret the factors, the component matrix is rotated using the Varimax method and Kaiser normalization (Kaiser 1958).

An item is assigned to a factor according to its factor load. The factor load is calculated using the correlation of a variable to the factor that was extracted from the data. In order for an item to be assigned to a design factor, the respective factor load has to be the highest value for this item and also has to exceed 0.5 (Härdle and Simar 2003).

Table 1 Rotated component matrix. (Saat et al. 2011)

		Factor		
Cat.	Item	1	2	3
BUS	Decision support	0.753987	0.211965	0.188453
ITG	Acquire and implement	0.714991	0.186588	0.225826
ITG	Plan and organize	0.677433	0.224612	0.277512
BUS	Control and follow up	0.674663	0.096948	0.268977
BUS	Integration and coordination	0.637061	0.279205	−0.019899
BUS	Flexibility	0.636681	0.24160	−0.235255
BUS	Effectiveness	0.549010	0.483476	0.04557
ITS	Performance	0.051344	0.776337	0.196293
ITS	Availability	0.127147	0.69881	0.167507
ITS	Suitability	0.318333	0.654633	0.142230
ITS	Usability	0.275959	0.651576	−0.073737
ITS	Accuracy	0.184559	0.632516	0.338447
BUS	Efficiency	0.512070	0.537886	−0.002942
ITS	Maintainability	0.417643	0.516266	0.120234
ITS	Security	−0.056182	0.159936	0.768478
ITG	Deliver and support	0.373640	0.321855	0.631850
ITG	Monitor and evaluate	0.531410	0.061938	0.620340

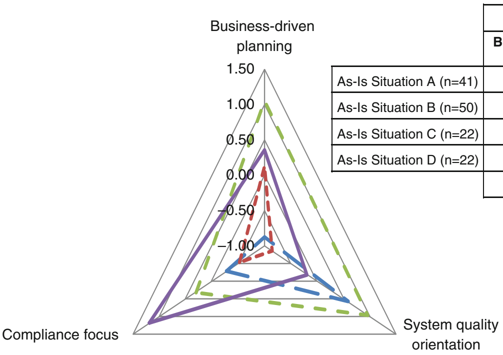
BUS Business quality, *ITS* IT systems quality, *ITG* IT governance quality

Design factor 1 *Business-driven planning* consists of business and IT governance concerns. Items such as decision support, control and follow-up, business flexibility, effectiveness, integration and coordination, acquire and implement as well as plan and organize load on this factor. Participants (and hence IT/business alignment problems) scoring high in this factor emphasize strong business orientation and a strong IT organization.

Design factor 2 *System quality orientation* subsumes high priority of performance, availability, usability, accuracy, maintainability and suitability of the used IT systems. Therefore, companies (and hence IT/business alignment problems) scoring high in this factor focus on delivery of high-class and effective delivery of IT services to users.

Design factor 3 *Compliance focus* consists of the three items systems security, monitor and evaluate, and deliver and support. Common characteristics can be found in striving for a high degree of security and continuity of systems and business processes.

In order to identify as-is design situations within the data set, hierarchical cluster analysis is applied to the factor values calculated above. The fusion algorithm used is the Ward algorithm using Eucledian squared distance. This course of analysis creates distinct clusters by minimizing the internal cluster variance. This is the most popular approach for hierarchical clustering (Hair et al. 2006). The analysis results in four distinct design situations as presented in Fig. 4. The table presents the factor load for each cluster and the web diagram shows a graphical representation of these values.



	Factor 1 Business-driven planning	Factor 2 System quality orientation	Factor 3 Compliance focus
As-Is Situation A (n=41)	–0.88	0.58	–0.27
As-Is Situation B (n=50)	0.12	–0.86	–0.53
As-Is Situation C (n=22)	1.05	0.95	0.31
As-Is Situation D (n=22)	0.36	–0.19	1.19
	arithmetic means of factor scores for each cluster		

Fig. 4 IT/business alignment design situations. (Saat et al. 2011)

5 Artefact Design: Four IT/Business Alignment Solutions

In the preceding chapter, a semantic interpretation of the three design factors (step 8) has already been provided:

1. *Design factor 1 "Business-driven planning"* means that there is a strong (or weak) business orientation and a strong (or weak) IT organization.
2. *Design factor 2 "System quality orientation"* means that there is a strong (or weak) focus on delivery of high-class and effective IT service delivery to users.
3. *Design factor 3 "Compliance focus"* means that there is a high (or low) degree of security and continuity of systems and business processes.

The four identified IT/business alignment design situations within the analyzed sample can be then interpreted as follows:

- *Design situation A: technical quality biased*
  The first situation scores highest on the factor *system quality orientation*. Business planning issues are not regarded as very relevant. The degree of IT/business alignment is regarded to be rather low.
- *Design situation B: business demand biased*
  Business-driven planning is the most important issue for situation B. Technical quality is regarded much less relevant.
- *Design situation C: aligned innovation biased*
  Situation C delivers equally high scores for business and the IT-driven dimensions, while also considerably incorporating compliance aspects. It can be considered as a rather advanced group where a significant amount of IT/business alignment problems have already been addressed.
- *Design situation D: compliance biased*
  This situation is quite compliance focused and also somewhat business oriented. It is characterized by high perceived IT/business alignment. IT quality issues are regarded not very relevant.

Step 9 of the proposed procedure brings design situations together with design factors that seem to be of particular interest. The following combinations are obvious:

- *Design situation A:* Design problems represented by cluster A have the worst scores for design factor 1 and medium scores for design factor 3. They score well for design factor 2. The most interesting improvement linkages are to design factors 1 and 3.
- *Design situation B:* Design problems represented by cluster B have the worst scores for design factor 2, bad scores for design factor 3 and medium high scores only for design factor 1. There seems to be a lot of room for improvement in this cluster 2. The most interesting improvement linkage seems to be design factor 2.
- *Design situation C:* Design problems represented by cluster C have high scores for both design factors 1 and 2; Their scores for design factor 3 are medium. There seems to be not too much to improve in this cluster 2. If at all, the most interesting improvement linkage is to design factor 3.
- *Design situation D:* Design problems represented by cluster D score by far highest for design factor 3. Their scores for design factors 1 and 2 are medium. If at all, the most interesting improvement linkages are to design factors 1 and 2.

The desired "movements" are:

- *Design situation A:* IT/business alignment can be enhanced by (a) a stronger focus on business, (b) a more professional IT organization and (c) more emphasis on security and continuity. These measures promise strong effects on IT/business alignment.
- *Design situation B:* IT/business alignment can be enhanced by better and more effective IT service delivery. This measure promises strong effects on IT/business alignment.
- *Design situation C:* Although both the business and the technical perspective are quite mature in situation C, IT/business alignment could be enhanced by an increased focus on security and continuity of systems and business processes. Effects will be small.
- *Design situation D:* While overarching issues like security and continuity have quite a high maturity in situation D, both (a) business focus, (b) the IT organization's professionalism and (c) IT service management could further be enhanced. Effects will be medium.

Step 10: Since all IT/business alignment qualities are regarded as equally important and desirable and because no explicit tradeoffs between qualities are evident, the "ideal solution" would be close to design situation C—maybe with a little more emphasis on overarching issues like security and continuity.

Since a comparatively small number of only three design factors has been identified in the sample, there is no need to aggregate different problem solution fragments into a complex problem solution (step 11). Based on the assessed effect of measures, the following solution artefacts would result from our analysis:

1. If IT/business alignment deficits result from a technical bias, mainly business qualities in the fields of decision support, acquire and implement, plan and organize, control and follow up, integration and coordination, flexibility and effectiveness need to be addressed. These activities should be accompanied by addressing IT governance qualities like security, delivery and support, as well as monitor and evaluate.
2. If IT/business alignment deficits result from a business demand bias, mainly IT qualities like performance, availability, suitability, usability, accuracy, efficiency and maintainability should be addressed.

The scales used for the characterization of the respective items in the questionnaire provide additional hints on how “improvement” or “enhancement” can be operationalized by respective solution artefacts.

6 Discussion and Outlook

In the preceding chapters, we have shown how problem analysis data can be used to systematically explore a design problem class, identify design factors and design situations, and transform these insights into solution artefact design. Based on the outlined artefact design, generic solutions like reference models and methods can be constructed and subsequently adapted to specific problems. While this procedure is based on survey data of related design problems in organizations, it can be applied to any problem within the addressed design problem class. For every design problem, the “closest” design situation of that class can be identified, and a solution is then based on the respective generic solution artefact.

A survey of IT/business alignment problems has been used to illustrate the proposed approach. While this problem class proved suitable to obtain a sufficiently sized data set which moreover showed very good sampling quality for the applied instruments, only three design factors were identified. As a consequence, the four discovered design situations could only be associated with comparably simple and straightforward design fragments, and the resulting aggregate solution artefact design was much simpler than in other design problem classes. For other design problem classes, we observed six to eight design factors and more complex solution artefacts.

The illustrative survey example also did not exhibit a differentiation between as-is problem data and to-be goal data. If as-is as well as to-be data are collected, different types of situations (as-is vs. to-be) can be discovered in parallel, and much better targeted design activities / project types can be identified.

Another simplification of the used application example is the lack of tradeoffs between design goals and design activities: Since all qualities should be achieved equally, the derivation of “movement” fragments and their aggregation into situational artefacts was straightforward. If tradeoffs have to be observed, both the fragment specification and the fragment aggregation become much more complex.

An interesting feature of many design problem analyses that yield a larger number of design factors is that the first factor is often representing many and quite diverse problem aspects that are sometimes not easy to interpret semantically. With regard to design problem analysis and solution construction, we interpret this “technically” overloaded factor as a problem independent aggregation of “generalized” properties and the respective solution fragment as a basic set of domain-independent problem solution activities like e.g., general project/transformation management. This aspect of our approach does certainly need additional research effort.

In addition, our approach does not explicitly cover yet the adaptation of situated solution artefacts to specific design problems. On the one hand, we consider this extension not too problematic because there is a plethora of adaptation knowledge on reference models which promises to be generalizable to generic artefacts (including methods). On the other hand, adaptation efforts might depend on problem properties and influence the “optimal” level of artefact genericity that up to now is determined using “technical” homogeneity/heterogeneity metrics only.

Another and probably the most important extension would be to include not only adaptation effort, but also other “economical” properties like the number of problem instantiations of a type or the attractiveness of problems in terms of economic gains into the identification procedure of design factors and in particular design situations.

Acknowledgements The IT/business alignment survey has been designed, conducted and interpreted together with Jan Saat, University of St. Gallen, as well as Ulrik Franke, Robert Lagerström and Mathias Ekstedt, all from KTH Stockholm.

The author wishes to thank Christian Fischer and Felix Wortmann, both University of St. Gallen, for fruitful discussions on constructional modelling methodology and for comments on this paper and its predecessor.

References

Aier S, Winter R (2009) Virtual decoupling for IT/business alignment—conceptual foundations, architecture design and implementation example. Bus Inf Syst Eng 51(2):150–163

Baskerville RL, Pries-Heje J et al (2009) Soft design science methodolog. In: Proceedings of the 4th international conference on design science research in information systems and technology (DESRIST 2009), ACM, New York

Baumöl U (2006) Methodenkonstruktion für das Business-IT-Alignment. Wirtschaftsinformatik 48(5):314–322

Becker J, Delfmann P et al (2002) Konfigurative Referenzmodellierung. In: Becker J, Knackstedt R (eds) Wissensmanagement mit Referenzmodellen. Konzepte für die Anwendungssystem- und Organisationsgestaltung. Physica, Heidelberg, pp 25–144

Becker J, Delfmann P et al (2007) Adaptive reference modeling: integrating configurative and generic adaptation techniques for information models. In: Becker J, Delfmann P (eds) Reference modeling. Physica, Heidelberg, pp 27–58

Bucher T, Klesse M (2006) Contextual method engineering. Institute of Information Management, University of St. Gallen, St. Gallen

Bucher T, Klesse M et al (2007) Situational method engineering—on the differentiation of “context" and “project type". Situational method engineering—fundamentals and experiences. Springer, Boston

Cattell RB (1966) The scree test for the number of factors. Multivariate Behavioral Res 1(1):245–276

Gammelgård M, Ekstedt M et al (2006) A categorization of benefits from IS/IT investments. In: 13th European conference on information technology evaluation (ECITE 2006), Academic Conferences International, Genoa

Gustafsson P, Franke U et al (2008) Quantifying IT impacts on organizational structure and business value with extended influence diagrams. In: The practice of enterprise modeling (PoEM 2008). Springer, Stockholm

Hair JF Jr, Black B et al (2006) Multivariate data analysis. Prentice-Hall, Englewood Cliffs

Härdle W, Simar L (2003) Applied multivariate statistical analysis. Springer, Berlin

Henderson JC, Venkatraman N (1999) Strategic alignment: leveraging information technology for transforming organizations. IBM Syst J 38(2/3):472–484

Hevner AR, March ST et al (2004) Design science in information systems research. MIS Q 28(1):75–105

Johnson P, Ekstedt M (2007) Enterprise architecture: models and analyses for information systems decision making. Studentlitteratur, Pozkal

Kaiser HF (1958) The varimax criterion for analytic rotation in factor analysis. Psychometrika 23(3):187–200

Kaiser HF, Dickman KW (1959) Analytic determination of common factors. Am Psychol Rep 14:425–430

Kaiser HF, Rice J (1974) Little Jiffy, Mark Iv. Educ Psychol Meas 34(1):111–117

Luftman JN (2000) Assessing business-IT alignment maturity. Commun AIS 4(14):1–50

Luftman JN (2005) Key issues for IT executives 2004. MISQ Executive 4(2):269–285

Luftman JN, Kempaiah R (2008) Key issues for IT executives 2007. MISQ Executive 7(2):99–112

Luftman JN, Kempaiah R et al (2006) Key issues for IT executives 2005. MISQ Executive 5(2): 81–99

Luftman JN, Kempaiah R et al (2009) Key issues for IT executives 2008. MISQ Executive 8(3):151–159

March ST, Smith GF (1995) Design and natural science research on information technology. DSS 15(4):251–266

Plazaola L, Flores J et al (2007) An approach to associate strategic business-IT alignment assessment to enterprise architecture. In: 5th conference on systems engineering 2007 (CSER2007), Stevens Institute of Technology, Hoboken

Saat J, Winter R, Franke U, Lagerström R, Ekstedt M (2011) Analysis of IT/business alignment situations as a precondition for the design and engineering of situated IT/business alignment solutions. 44th Hawaii international conference on System Sciences (HICSS-44), Kauai, HI

Sledgianowski D, Luftman JN (2005) IT-business strategic alignment maturity: a case study. J Cases Inf Technol 7(2):102–120

Thompson BJ (2004) Exploratory and confirmatory factor analysis: understanding concepts and applications. American Psychological Association, Washington

Winter R (2011) Problem analysis for situational artefact construction in information systems. In: Carugati A, Rossignoli C (eds) Emerging themes in information systems and organization studies. Physica, Heidelberg, pp 97–113

Zachman JA (1987) A framework for information systems architecture. IBM Syst J 26(3):276–292

Preferences of Business Process Models: Interpretative Analysis on Spontaneously Common Symbols

Saleh Al-wahaishi, Ahmad Jaffar, Ivo Vondrák, Václav Snášel and Jaroslav Pokorný

Abstract Through quantitative analysis, previous researches had proven a significant preference towards a specific set of notations for modeling business processes. The drawn conclusion revealed a significantly correlated coefficient preference to Norm Process Chart for using easily recognizable symbols to intuitively elicit clear understanding in representing business process models. Further interpretative analysis to qualitatively enhance these findings will only prove and strengthen the above claimed beyond reasonable doubt. The approach is to measure respondent level of accuracy in interpreting 3 different case studies modeled using 3 different modeling techniques shown to respondents in three different randomized sequences. The analysis includes correlating the finding against the time taken as well as respondents' level of confidence in interpreting these models. The significantly correlated results again confirmed beyond reasonable doubt Norm Process Chart being respondents ultimate choice. Further comparative analysis between results from an earlier investigation against the latter, revealed similar patterns in respondents' responses despite respondents dispersed ethnicity and educational backgrounds.

1 Introduction

Business Process Modeling (BPM), an approach to graphically display the way organizations conduct their business processes, has emerged as an important and relevant domain of conceptual modeling. It is considered as a key instrument for the analysis and design of process-aware Information Systems [5].

BPM advocates the use of symbolic notations to represent business processes. Influenced by system engineering and mathematics, the application of these notations involves technical processes designed by engineers, undertaken by technically trained analysts for the use of largely technical people [5]. However, majority of business

J. Pokorný (✉)
Charles University of Prague, Prague, Czech Republic
e-mail: pokorny@ksi.mff.cuni.cz

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_4,

process stakeholders are non-technically inclined with business or administrative background. While some notations are comprehensive, their arbitrary geometrical symbols can be visually and technically complex; and cognitively difficult to understand with unclear semantics. While such representational constraints prevent effective communication of process knowledge, initial hypothesis proving investigation [6] substantiated that Norm Process Chart (NPC) [5], a newly proposed set of notations for modeling business processes, is semantically clearer than existing BPM notations. Using various statistical analysis techniques [2, 4] the investigation revealed a linear relationship among all the variables used for this comparative analysis. With its correlation coefficient significantly at 0.1 level, not only the various results eliminate possible chance of bias but also highlighted significant pattern in favor of NPC with the mean scores (asymptotic significance at 0.000 for its variable grouping) for other techniques almost double as compared to NPC [6].

However, the above investigation falls short at measuring confidently the level of accuracy in respondents' interpretation of the given models. This generates a subsequent hypothesis that respondents are able to interpret processes represented using NPC more accurately than using other notations. Analysis will be based on experimental data collected through survey using different sets of respondents from the initial investigation. Using a free-format answer, respondents will be asked specific questions in relation to their interpretation of three process fragments, which will be represented by the three different notations in three different sequences. Respondents' answers will be analyzed to measure the level of accuracy in terms of their understandings of the given model, their level of confidence as well as time taken to appraise the given models. Detailed considerations about various properties and usability of visual techniques can be found in paper [9].

This paper presents an experiment designed to substantiate the developed hypothesis objectively by comparing NPC with two well-established approaches—Integrated Definition [1, 8] and Roles Activity Diagrams [3, 10]. In Sect. 2 we shortly repeat principles of notions considered and the design of questionnaire. Sections 3 and 4 describe the experiment and present the results. Future work based on analyses of the outcome is also indicated in Sect. 5.

2 Designing an Interpretative Survey

An experiment was proposed with the aim of comparing which notation is accurately interpreting a given model. The term accurate interpretation determines the number of correct answers to questions in interpreting a given model for representing business processes. The hypothesis asserts that there is a significant accuracy in interpretation for one notation; its opposite asserts otherwise.

Using a conclusive research technique [7], in the form of a questionnaire survey, respondents were asked to interpret the different process representations using **N**orm **P**rocess **C**hart (see Fig. 1), **R**ole **A**ctivity **D**iagram (see Fig. 3), and **I**ntegrated

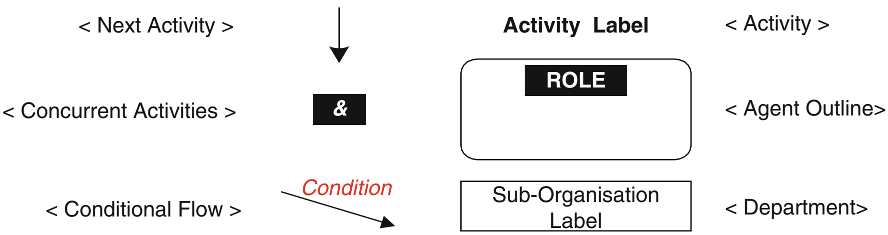


Fig. 1 NPC basic symbols [5]

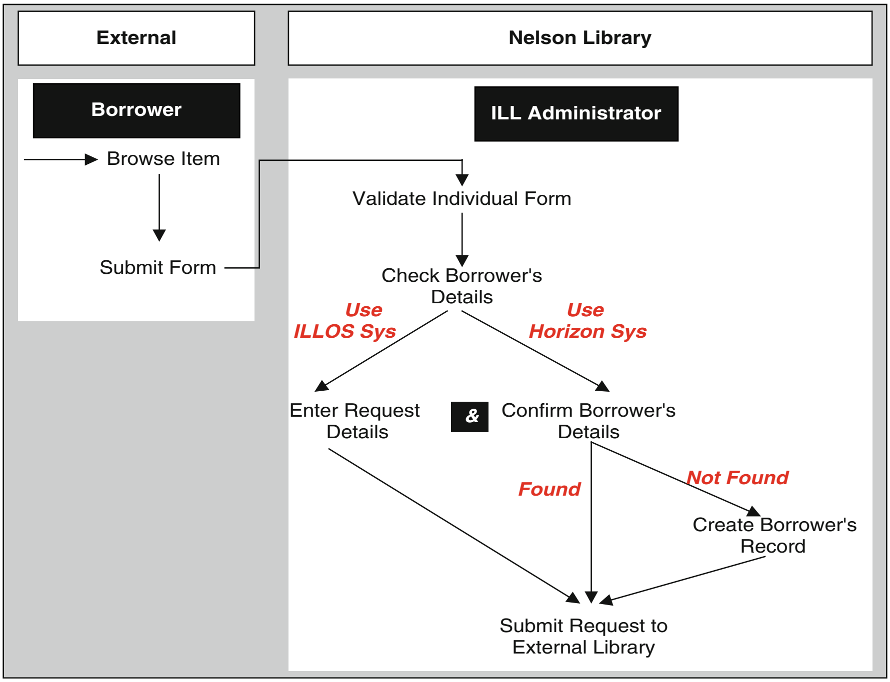


Fig. 2 Inter-Library loan using NPC

DEFinition (see Fig. 5). These representations were applied to different process extractions (Insurance Claim, PhD Registration, and Inter-Library Loan) in different randomized sequence based on Latin Square technique [2, 1]. The latter produced randomized questionnaires with 36 sets of different combinations. Respondents should answer various questions for the purpose of identifying the accurate understanding of these models presented to them. The contention was to determine whether accurate interpretation for one notation in modeling business process is much higher as compared to another.

Similar design for the questionnaire [6] was adopted to maintain consistency. The intention here is to elicit similar behavioral pattern in the statistical analysis.

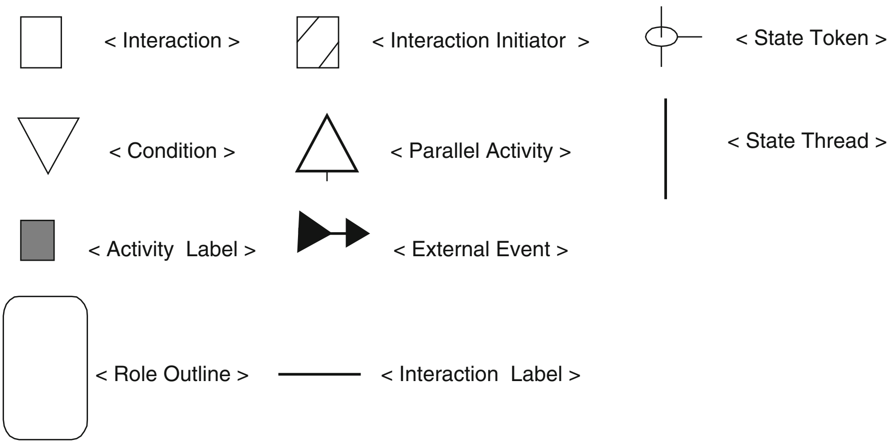


Fig. 3 RAD basic symbols [10]

The proposed questions were directly testing respondents' accuracy in interpretation based on their understanding of the process flow and logic for the given models. Respondents must accurately list the various activities in its correct sequence, identifying conditional rules and parallel activities used within the process. They must also name the various participating agents and identify who triggers their interactions and finally determine their level of confidence in their answer using a Likert Scaling technique [11].

3 Results in Analyzing Interpretative Survey

The analysis must reveal a much higher level of interpretation accuracy for one notation as compared to others (NPC, RAD or IDEF). Despite a low respond of 33% returned questionnaires, the various analyses depicted a very strong consistently significant correlated inclination towards a particular set of notation in modeling business processes. Majority of the respondents interestingly were undergraduates with minimal exposure to organizational processes and understanding on modeling techniques.

Eliminating bias results from the findings, various variables were compared against one another to assess significant correlation. The analysis compared time taken to do the questionnaire against the overall level of accuracy in interpreting the model. It also compared the latter to the level of confidence to determine respondents' state of well-being while interpreting the models. In substantiating the accurate interpretation versus level of confidence, a detail analysis on two specific questions was also being solicited.

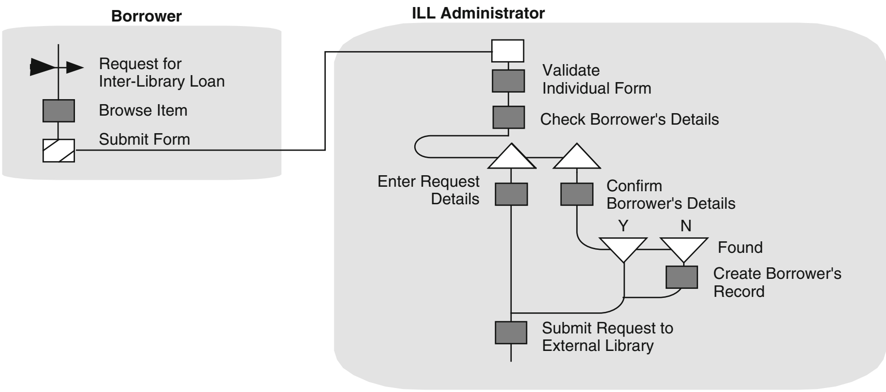


Fig. 4 Inter-Library loan represented using RAD

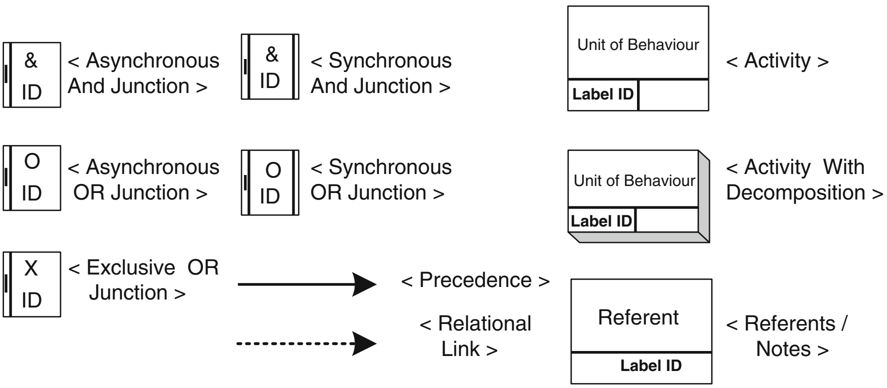


Fig. 5 IDEF suite basic symbols [8]

3.1 Results: Time vs. Accurate Interpretation

With reference to Fig. 7, 42% of the respondents solved the NPC model questions within five minutes with 71% of them scored between 80 and 100% (see Fig. 8). As compared to RAD model, 71% of respondents took between 6 and 15 minutes to solve the model questions with only 39% of them scored within the 80–100%.

Despite 16% of the respondents managed to complete the questionnaire within 5 minutes, IDEF fair worse with only 3% of respondents were able to interpret and answer the various questions relating to the model accurately. The majority had great difficulties in appreciating and interpreting the various geometrical notations used to model the processes. Upon correlating time taken to solve the models' questions to the accurate interpretation, NPC model outclass others, followed RAD and finally IDEF models.

3.2 Results: Accuracy vs. Respondents' Confidence

As in Fig. 9, respondents' level of confidence reflected 48% of them were 75–100% confident in solving the NPC model. Whereas only 36 and 3% expressed the same level of confidence when solving RAD and IDEF models, respectively. Correlating this result against the findings on accurate interpretation, it's matched perfectly for all the various models with NPC attaining 71% of respondents achieving more than 80% accuracy in their interpretation. Similarly to RAD and IDEF respectively, with the 45% majority in RAD attained 50–70% accuracy and 68% majority in IDEF managed the maximum of 40% accuracy.

The correlation between these 3 variables (time taken, accurate interpretation, and level of confidence) thus far reflected significantly consistent in pattern in preference to NPC. Not only NPC has the highest majority in the shortest time to answer the questionnaire, its differences of 71% in level of accuracy almost doubled the percentage attained by RAD and IDEF combined together. Similarly, if 50% is the benchmark for level of confidence, NPC with 84% respondents attained about 20% more majorities to RAD at 68% against IDEF at 35% only. RAD as compared to IDEF held in second place consistently through.

3.3 In-depth Analysis on Specific Variables

This analysis focused on accurately interpreting the notations for representing start/end of process as well as accurately determining the correct sequence of activities flow. The contention is to substantiate the analysis on overall achievement attained by the above various variables. The results depicted significantly correlated patterns in favor of NPC. Unlike IDEF, majority of respondents were able to attained accuracy level of 75% or more in identifying the notation for start/end of process. NPC with the majority of 93% depicted itself as being the easiest to be identified as compared to RAD managing at 62% only. On the contrary, Fig. 11 highlighted an opposite pattern with 59% majority attained more than 75% accuracy for NPC. The RAD and IDEF majority only managed a maximum of 25% accuracy in interpreting the sequence of activity flow. RAD 29% of respondents attained above 75% accuracy while IDEF had only 7%. Nevertheless, results revealed that significantly correlated consistency attributed to the various analysis in Sects. 3.1 and 3.2.

4 Respondents' Preferences

While previous investigation proved that NPC is much preferred notations for representing business model, the main contention here is to prove that respondents can accurately interpret processes represented using NPC more than other notations. The above analysis had proven beyond reasonable doubts that the hypothesis is true.

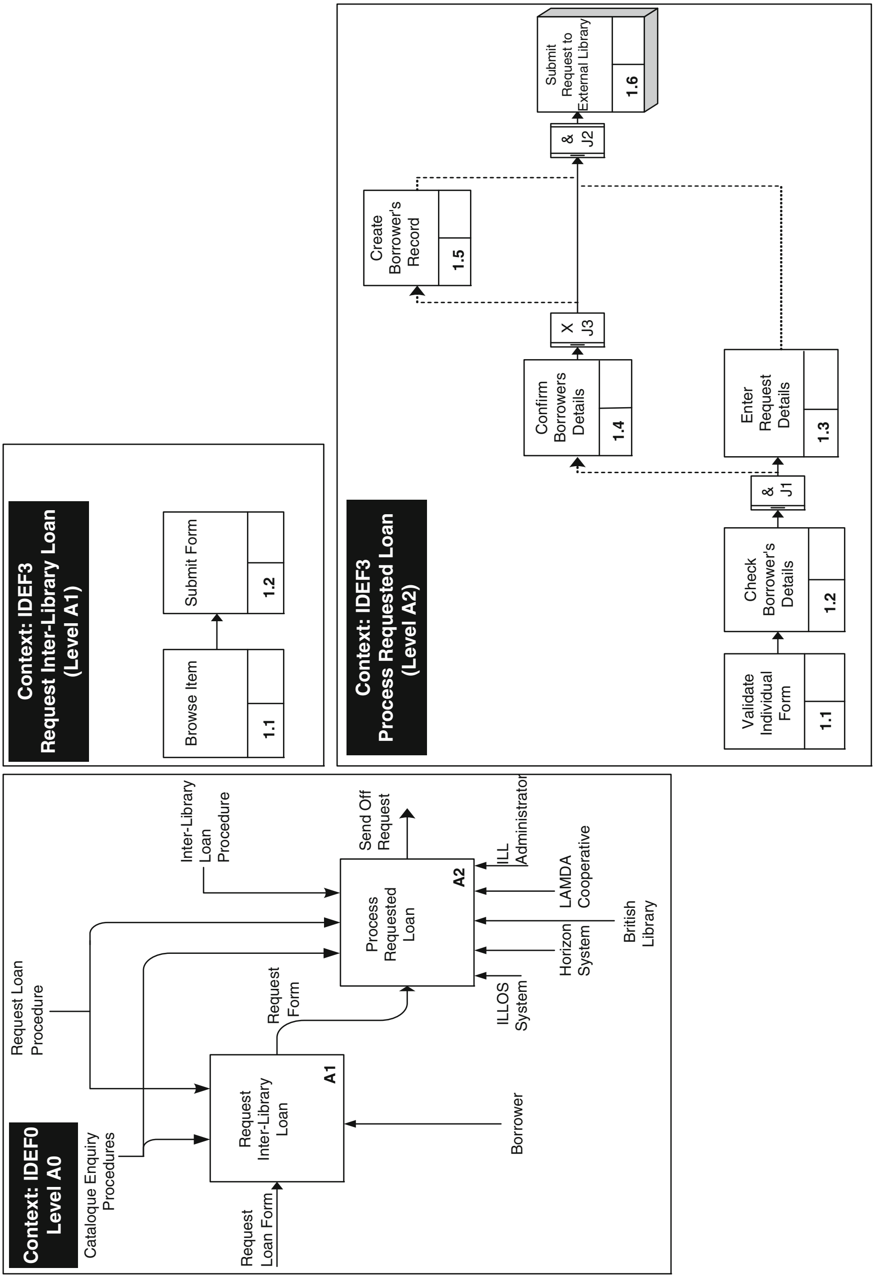


Fig. 6 Inter-Library loan represented using IDEF suite

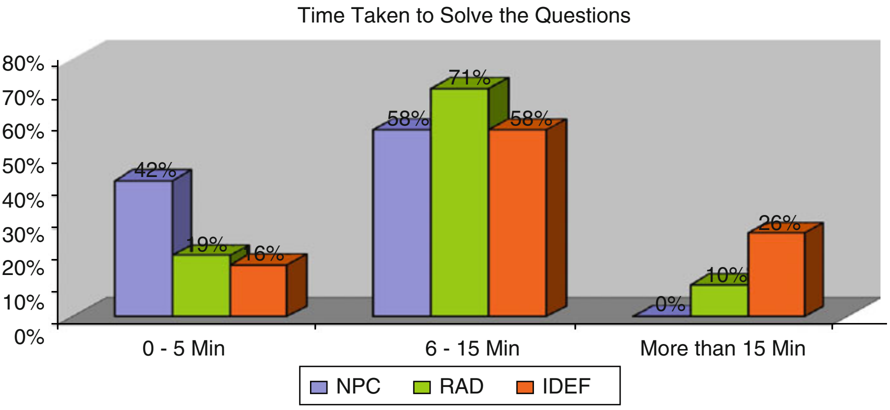


Fig. 7 Time taken solving questions for each model

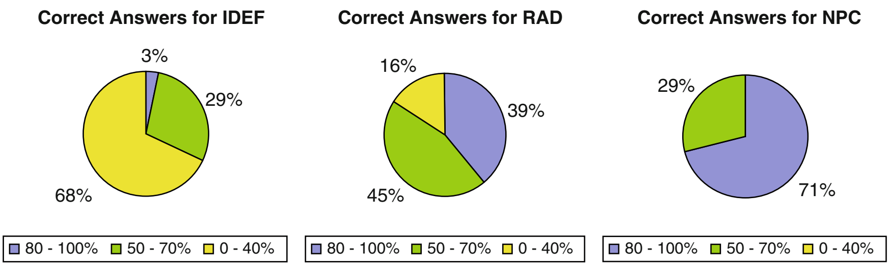


Fig. 8 Accurate interpretation for each method

Even with a quick glance on Fig. 2 as compared to Figs. 4 and 6 for similar process fragment, this hypothesis stands.

Upon cross-referencing each variable used for analysis, the above overwhelmingly revealed significantly correlated relationships between each of them in favor of NPC. While respondents not only took less time to appreciate and interpret NPC model for all type of process fragments in randomized sequence, their accuracy in interpretation that reflected their level of understanding of what the model presented were equally high. This is clearly confirmed with their high level of confidence in answering the questionnaire for NPC. IDEF experienced the brunt of respondents' displeasure for its overwhelmingly technical inclination towards its representation. Not only have most respondents unable to correctly interpret the meanings of its geometrical notations, the various levels of abstract representation in itself confused respondents more so that RAD. Unlike NPC and IDEF being in the extreme of the equation, on the contrary, RAD overall average balances in the middle for almost all of the analysis. It reflected upon certain level of complexity within its model, where respondent have some difficulties but not overly complex as with IDEF.

In comparing the findings of this investigation against the previous (see Figs. 10, 11 and 12), similar pattern consistency towards NPC was very evidence. All

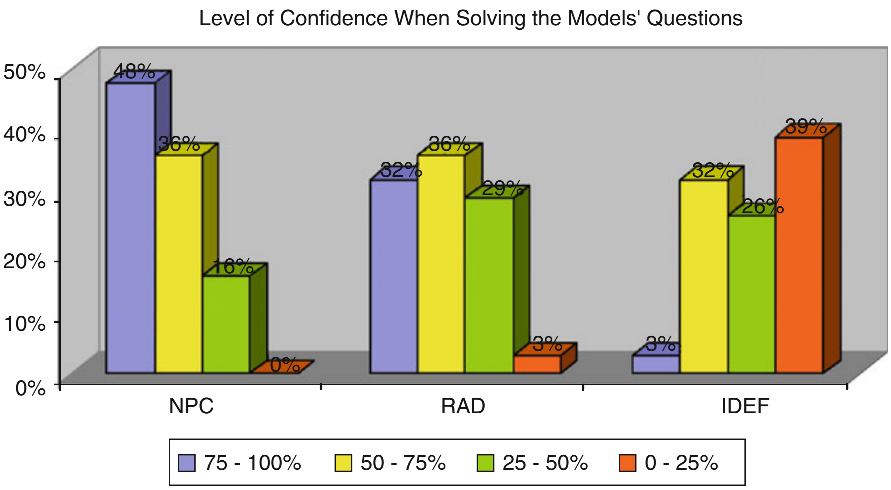
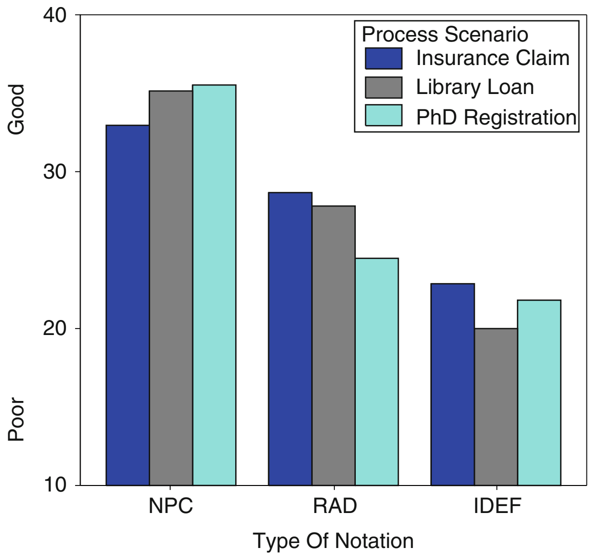


Fig. 9 Confidence level for each method

Fig. 10 Overall scores by type of process fragments



analyzed variables shown significant preference towards NPC, followed by RAD and subsequently IDEF, which mapped directly to those experienced in this analysis. It's further proved that both hypotheses where NPC is much preferred for its readability and semantically much clearer; it is also accurately interpreted with highest level of confidence within the shortest time frame.

The analysis did not reveal constraint in dependence on English being the medium of communication. On the issue of ethnicity, working and educational background,

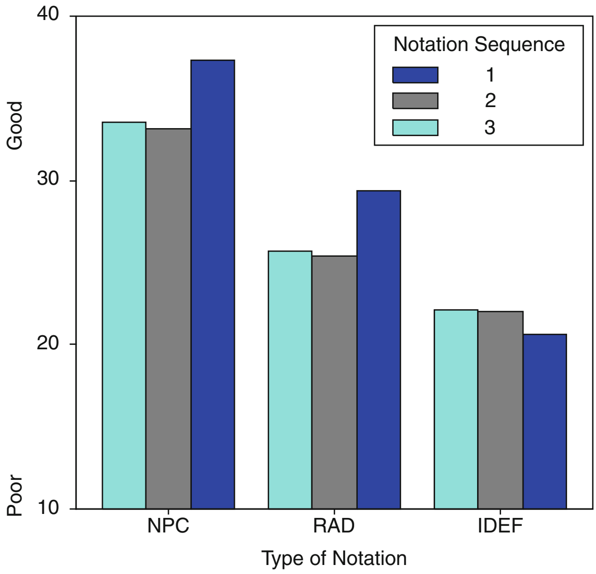


Fig. 11 Overall scores by notation sequence

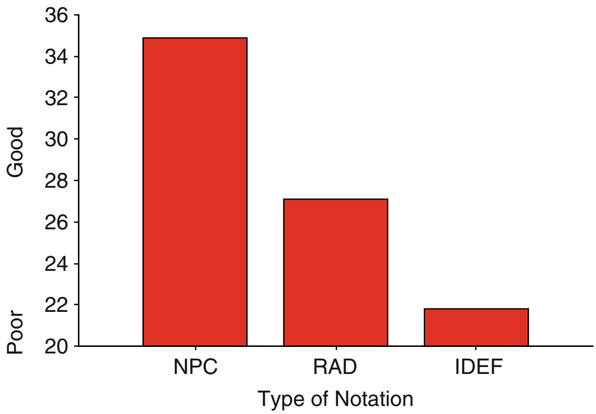


Fig. 12 Overall scores by type of notation

where the majority of the respondents are native Arabic mostly educated in an Arabic state school, NPC projected as being more language friendly with universally common notations of arrow and statement controls as compared to RAD and worse IDEF. Similarly, NPC also could be projected as being less technically inclined. This was derived from the fact that majority of respondents were fresh undergraduates neither with much organizational and business process experiences nor exposure to modeling technique except for Dataflow Diagram as part of their elementary undergraduate studies. The fact that they formed the bulk of respondents and yet achieved high percentage of accurate interpretation confidently within a short time frame, highlighted the unnecessary technical exposure needed to decipher NPC models. Ironically, RAD and IDEF being more technically influenced attained reversed scores against them as compared to NPC.

5 Conclusion

This research hypothesizes that certain notation can be interpreted much more accurately as compared to other notations. It has also proved that there exists pattern consistency in its findings with a previous investigation [6] and that ethnicity and educational background has no bearing on its outcome. The hypothesis had been proven in favor of NPC both in terms of quantitative evidences from the previous investigation as well as the above interpretative research findings. Not only NPC co-efficiently and significantly correlated in respondents' preferences in the initial investigation but also proven to be accurately interpreted by different groups of dispersed origin and educational backgrounds as in the above analysis. Unlike the correlated variables of accurate interpretation versus time and level of confidence, there were weak comparative analysis on proving the pattern consistency and respondents' background. Nevertheless, NPC had been accorded with being the most preferred as well as easily and accurately interpreted as compared to RAD and IDEF.

Other analysis method must be advocated to address the identified weak comparative analysis. While the respondents' population is subject to scrutiny, extending the survey to include various industrial partners will strengthen the findings into proving NPC usability, readability and perhaps weaknesses as well. In discovering such favorable results towards NPC prompt further explanation on its achievement. With the use of developed tools such as GATE within the realm of Natural Language Processing will be useful to further emphasize the consistency in accurate interpretation via techniques such as text parsing and pattern matching in relation to respondents' textual answers to various survey questions.

Acknowledgements This research has been partially supported by the grant of GACR No. P202/10/0761.

References

1. Bose R, Manvel B (1984) Introduction to combinatorial theory. Wiley, New York
2. Carter D (1997) Doing quantitative psychological research: from design to report. Psychology Press, Hove
3. Holt A, Ramsey H, Grimes J (1983) Coordination system technology as the basis for a programming environment. ITT Tech J (Electr Commun) 57(4):307–314
4. Howitt D, Cramer D (1999) A guide to computing statistics with SPSS for Windows—Version 8. Prentice-Hall, Englewood Cliffs
5. Indulska M, Recker J, Rosemann M, Green P (2009) Business process modeling: current issues and future challenges. In: van Eck P, Gordijn J, Wieringa R (eds) Advanced information systems engineering—CAiSE 2009, LNCS, vol 5565. Springer, Amsterdam, pp 501–514

6. Jaffar A, Rowland H, Alderson A, Flecther M (2003) An empirical comparison of some business processes notations. In: Arabnia HR (ed) International conference on information and knowledge engineering—IKE'03, CSREA Press, Las Vegas, pp 344–350
7. Joppe M (2010) The research process. http://www.htm.uoguelph.ca/MJResearch/ResearchProcess/default.html. Accessed Mar 2010
8. Knowledge Based Systems, Inc. (2010) A structured approach to enterprise modeling and analysis. http://www.idef.com/default.html. Accessed Mar 2010
9. Moody DL (2009) The "physics" of notations: toward a scientific basis for constructing visual notations in software engineering. IEEE Trans Softw Eng 35(6):756–779
10. Ould M (1995) Business process modelling and analysis for reengineering and improvement. Wiley, New York
11. Research Methods Knowledge Base (2010) Likert scaling. http://www.socialresearchmethods.net/kb/scallik.php. Accessed Mar 2010

Verification of *i** Models Using Alloy

Peter Oluoch Ating'a and Aneesh Krishna

Abstract Many modelling techniques focus on the later stages of requirements engineering and the design phase while not paying adequate attention to the earlier phases of requirements engineering. The *i** framework is a conceptual modelling language, which can be used to model an information system and its bounding environment together with the rationales for a particular system-environment configuration. However, the *i** framework has limited tool support for analysis. Alloy on the other hand is a modelling language that has tool support for automatic analysis. Combining the strengths of these two frameworks from the early stages of software development will provide better verification and validation mechanism. This paper presents the model transformation approach from *i** to Alloy and demonstrates the synergy between these two frameworks by way of an example of a meeting scheduler.

1 Introduction

The *i** modelling framework (Yu 1995) was developed for reasoning about the computer systems within an organisation. The framework takes into consideration the environment together with the stakeholders (and their 'mental states') of the proposed information system. The stakeholders have strategic dependency relationships with each other that can be analysed using a qualitative reasoning approach. However, these networks of dependency relationships have the potential to become large and complicated and there is limited tool support for the analyses of such networks (Krishna et al. 2009).

Alloy, on the other hand, is a software modelling language with a simple syntax that comes with tool support for model analysis. It is considered to be a light weight, formal method in the sense that it provides the benefits of traditional formal methods albeit at a lower cost.

A. Krishna (✉)
Department of Computing, Curtin University of Technology, Perth, Australia
e-mail: a.krishna@curtin.edu.au

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_5,

This paper aims to explore a potential synergy between the Alloy modelling language and the *i** framework, due to its simplicity and automated analysis in the modelling of computer systems. This paper will propose an Alloy model that is the result of the transformation of *i** modelling elements at a metamodel level into an Alloy model. It is intended that this model can be imported into future Alloy models to provide them with the expressive power of *i** modelling, with the additional benefit of automated model analysis.

This paper is organised as follows: Sect. 2 (model transformation) documents the transformation of *i** elements into an Alloy model. Section 3 (analysis) then analyses the transformation process and the resulting model. Section 4 (conclusion) then offers a brief summary and discusses potential future work regarding.

2 Model Transformation: *i** to Alloy

2.1 i Concepts*

The concepts central to the *i** framework includes: the intentional actor, dependums, external dependency relationships between actors, and internal intentional relationships within actors. The concepts are covered within the two models contained within the framework—the Strategic Dependency (SD) model, and the Strategic Rationale (SR) model. The SD model describes dependency relationships between various stakeholders (actors) at the organisational level, whereas the SR model describes the stakeholder interests.

In the *i** framework, an intentional actor possessing properties such as goals, beliefs, abilities and commitments. Actors represent the active entities that carry out actions to achieve goals. An inherent property of these intentional actors is the fact that they are strategic in nature. In order to achieve their goals and objectives they depend on other actors, and thus are concerned about opportunities and vulnerabilities associated with these dependencies. The actors thus seek to position themselves within the environment in a way that their interests would be best served.

The term dependum in the framework is a collective term for the elements that actors depend on each other for. These elements include goals to be achieved, tasks to be executed, resources to be provided and soft goals to be achieved (albeit a soft goal has no clear cut criteria for achievement). An in depth description of the framework is provided by Yu (1995).

2.1.1 Case Study

A case study model will be used as an example for the rest of the paper. The model used is that of the meeting scheduler (Yu 1997).

The proposed meeting scheduler requires that for each meeting request, it should select a location and date for which the most number of invited participants will

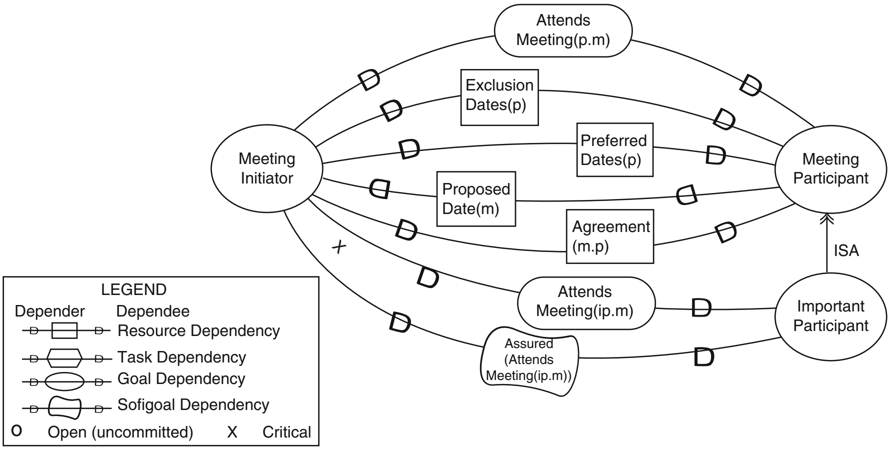


Fig. 1 SD model diagram for case study (Yu 1997)

attend. In this example, there are two actors: *Meeting initiator, Meeting participant* (and *Important participant)*. There are two goal dependums in this example: *Normal participant attends meeting* and *Important participant attends meeting*. There are four resource dependums: *Exclusion dates, Preferred date, Proposed dates* and *Agreement between initiator and participant*. Finally, there is soft goal dependum: *Assured meeting attendance*, between the meeting initiator and the important participant.

The SD model describing the relationships between the elements in this example is shown in Fig. 1.

This example contains an SR model for the meeting initiator and the meeting participants, which is depicted on the diagram (Fig. 2).

Upon examining the Meeting Initiator, this actor has a major internal task: *Organise meeting*. This task is decomposed into two soft goals: *Quick, Low Effort*; and a goal: *Meeting be scheduled*. The soft goals and the goal mentioned above are realised by a task: *Schedule meeting*. This indicates three means-ends links in which the two soft goals and are the ends, and the *Schedule meeting* task is the means.

Schedule meeting task is also further decomposed to two tasks: *Obtain Agreement, Obtain AvailDates*; and a goal: *Find Suitable Slot*. This goal is realised by a task: *Merge Available dates* suggesting another means-ends link between this goal and the aforementioned task.

2.2 Alloy Concepts

The main features of the Alloy (Jackson 2006) modelling language include signatures, fields, assertions, facts, functions and predicates.

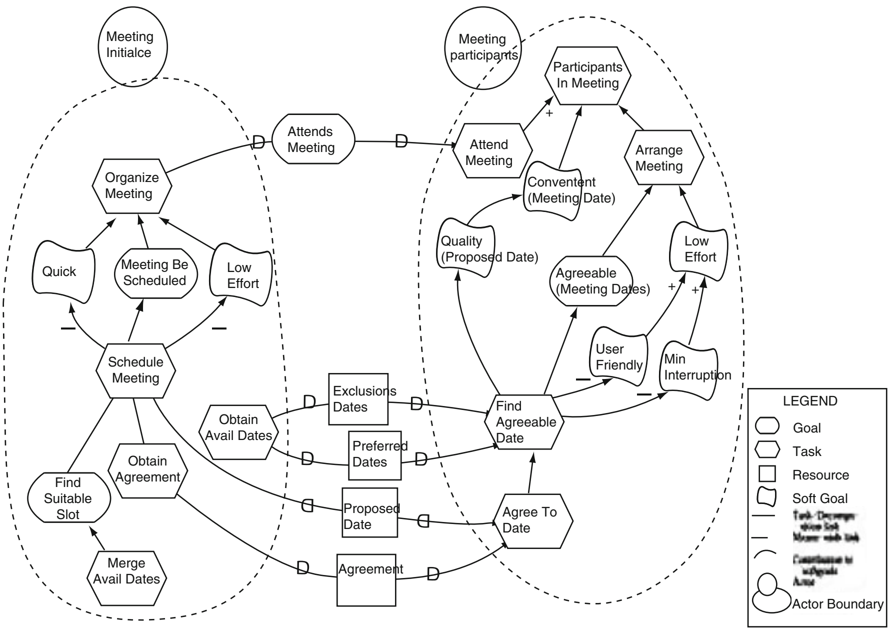


Fig. 2 SR model diagram for case study (Yu 1997)

A *signature* represents a data domain and its structure. Signatures represent 'types' in Alloy. Signatures may contain *fields*. A field represents a relation between a particular signature and other signatures. Signatures may also be extensions of other signatures to enable hierarchical specification.

Facts, functions and *predicates* represent various forms of constraints and expressions (Jackson 2006). *Facts* are explicit constraints on the model that are assumed to always hold. *Predicates* on the other hand are constraints on the model that are applied as required. Predicates represent reusable constraints. *Assertions*, on the other hand, represent properties over entire model that are expected to be true following from the facts within a model. *Functions* define reusable expressions.

The analyser (i.e. the model checker) operates by searching for examples to predicates that are being analysed, or by searching for counter examples to assertions that would indicate problems with the model created or the problem domain itself. The details of the Alloy are available from Jackson (2006).

2.3 i* SD Model Transformation

The SD model describes the intentional external dependency relationships between the actors within a particular environment or domain. The features of this model consist of a set of nodes that represent the actors (stakeholders) and the links between them. These links represent one actor depending on another for some element (Goal, Task, Resource, or Soft Goal), formally referred to as *dependum*. Thus in transforming the SD model, there are three key elements that need to be considered: actors, dependency relationships and dependums.

It is important to note that an actor can either be a *depender* or a *dependee*, depending on the particular *dependum*. However, the approach adopted in this paper is that an actor only ought to be aware of the *dependums* it requires and the *dependums* it provides, rather than the other actors that it has a relationship with. The issue of which actor is related to which other actor is handled by the properties of the *dependum*.

This suggests two top-level signatures: actor and *dependum.* These are shown below:

```
sig Actor{
provides: set Dependum,//represents the
          //dependums it provides and to who
requires: set Dependum// represents the
            //dependums an actor requires
}
abstract sig Dependum{
workability: Bool,//discussed in the following section
dependees: set Actor,//the actors providing the  //dependum
dependers: set Actor //the actors depending on the
//dependum
}
{      no actor:  Actor | actor in dependees &dependers    }
```

In lieu of the types of dependums i.e. Goals, Tasks, Soft Goals and Resources, there are four other signatures that extend the abstract 'dependum' signature above. These new signatures are shown below:

```
//GOAL
sig Goal extends Dependum {}
//RESOURCE
sig Resource extends Dependum {}
//SOFT GOAL
sig SoftGoal extends Dependum {}
//TASK
sig Task extends Dependum {}
```

The dependency relationships between actors is implied in the set of *dependums* that each actor requires or provides, since each *dependum* also has fields indicating which actors require it and which ones provide it. The reason for using this approach, as opposed to explicitly defining fields pertaining to the dependency relationships, is that this approach allowed an actor to be further classified implicitly as a *depender* or *dependee*, depending on the *dependum*.

2.4 i SR Model Transformation*

The SR model provides a more detailed description than the SD model in the sense that it describes internal intentional relationships between the elements of an actor and the *rationales* behind them. The approach used in the transformation of SR models to Alloy is aimed to enhance the SD Alloy model above to include the elements of the SR model, hence the Alloy models contained in this subsection will include expanded versions of the Alloy signatures above.

Like the aforementioned SD model, the SR model also consists of several types of nodes and links. The nodes in the SR model are based on the *dependum* types mentioned previously. There are two types of links between elements. The first, *means-ends* links, describe relationships between any of the *dependums* (the 'ends') and the way to attain them (the 'means'). These can be further classified depending on the type of *dependum*. However, the means is usually a task, and therefore in the Alloy model the means will always be tasks. The other type of link in the SR model is the *task decomposition* link, which as the name suggests, decomposes a task into its constituent elements that it requires in order for it to be completed.

Another key concept in SR models is the notion of *routines*, which represent a process and the rationales behind it. They are essentially a combination of elements that are the *means* to achieving some end, as opposed to a singular task to achieve that end. From this description it can be seen that a routine is essentially a type of means-ends link.

This suggests a new top-level Alloy signature, namely: *MeansEndsLinks*; which is an abstract signature from which the other *inter-dependum* relationships mentioned above inherit. These signatures are shown below:

```
/*The reason for having the means-ends links as explicit sig-
natures is to make it easier to characterise the concept of a
routine for the purposes of process analysis.*/
abstract sig MeansEndsLinks {}
sig GTLink extends MeansEndsLinks {
       end: Goal,
       means: Task
}
sig RTLink extends MeansEndsLinks {
       end: Resource,
       means: Task
```

```
}
sig TTLink extends MeansEndsLinks {
       end: Task,
      means: Task
}
sig STLink extends MeansEndsLinks {
       end: SoftGoal,
       means:Task,
       sgContribution: ContributionToSG
}
//ROUTINES
sig Routine extends MeansEndsLinks {
/*the assumtion is that a routine is defined from the end
goal downwards to the leaf nodes that provide the means to
achieve. The semantics behind the concept of a routine also
suggests an order of activities being executed  or 'ends' be-
ing reached.*/
end: Dependum,
means: Dependum,
subRoutine: Routine
}
{//the means in the routine should be the end of the
//subroutine
This.means == subRoutine.end
}
```

The *STLink* signature above has an extra field which suggests that an extra Alloy signature is required. This is specific to links where a Soft Goal is the 'end'. The importance for this stems from the fact that where there are alternative 'means' to achieve a Soft Goal, the *ContributionToSG* will determine which alternative to use for a particular routine.

```
//link contributions to soft goals
abstract sig ContributionToSG {}
sig Satisfisced, unsatisfisced extends ContributionToSG {}
```

Also based on the notions of routines, each actor can have a set of routines through which it can accomplish its objectives or realise goals. This suggests a modification to the actor signature from the SD model transformation discussed above to include routines. The signature thus becomes:

```
sig Actor {
provides:  set  Dependum,  //represents  the  dependums  it
//provides and to who
requires: set Dependum,// represents the dependums an //actor
requires
routines:  set  Routine  /*will  require  a  constraint  to
//ensure 'end' of the routines is in the 'provides' relation
*/
}
{
all routine_end: Dependum| routine_end in provides and  rou-
tine_end in routines.end
}
```

Based on the notion of task decomposition, the 'Task' Alloy signature is modified to include all the potential elements it could be decomposed to. The 'Task' signature thus becomes:

```
//TASK
sig Task extends Dependum{
//the possible nodes a task decomposes to
subTask: set Task,
subGoals: set Goal,
subResources: set Resource,
subSoftGoals: set SoftGoal
}
Fact {//no task is a sub task of itself
      no task:Task| task in  task.subTasks }
```

2.5 Routine Analysis Capability in the SR Model

The SR model also offers strategic analytical capabilities from the point of view of an actor in the context of a process (Routine). In particular, there are three main notions of reasoning in the current Alloy model performs: ability, workability and viability (Yu 1995).

Ability refers to whether an actor has a process (routine) to accomplish a particular goal. This however does not imply that an actor can achieve it alone (they may depend on other actors).

Workability refers to whether an actor believes they can successfully accomplish a process. A process is workable if all its elements are workable. In the Alloy model discussed here, the notion of workability is simplified to a Boolean field within the Dependum signature.

Viability refers to how well a process will work and is based on the *Contribution-ToSG* signature. A routine is considered to be viable if all its soft goals are *satisficed*. The notion on satisficing is same as discussed in Yu (1995). The proposed Alloy model implements these analytical concepts as predicates. These are shown below:

```
pred ability[actor: Actor, objective: Dependum]
{ objective in actor.provides }
```

The predicate above indicates that if a particular dependum is in the list of dependums an actor provides, then it has the ability to provide that dependum.

```
pred workability[actor: Actor, routine: Routine]
{
 all dependum: Dependum|
let routineEnds = (univ & Routine.end)|
dependum in routineEnds && dependum.workability==True  &&
routine in (univ & Routine)
}
```

The predicate above checks the *workability* field of all the elements of a particular routine, and if this field has a value of "True" in all the elements, then the routine is considered workable.

```
pred viability[actor: Actor, routine: Routine, links]
{
inRoutine in actor.routines &&
let stlinks =  (univ & STLink) |
{
       all r: Routine|
       r in ( inRoutine.^subRoutine) &&
       r.end in stlinks.end && stlinks.sgContribution = Sat-
       isfisced
}
}
```

The predicate above retrieves all the STLinks (Soft goal-task means-ends links) for a particular routine and examines the sgContribution field for each to check if they are 'Satisfisced' in order for the entire routine to be viable.

3 Analysis of the Meeting Scheduler Case Study

The case study *i** model was transformed to an Alloy model using the proposed guidelines. The first test run was performed for testing the notion of *ability*. A new predicate, *testAbility*, was defined that called the ability predicate from the initial

Fig. 3 Example model from testing the ability predicate on the case study model

transformation model. However, testAbility has as its parameters the *MeetingInitiator* and *OrganizeMeeting*, as opposed to general Actors and Dependums in ability. The results of running *testAbility* projected over only one actor command is shown in the Fig. 3.

The next test was performed for testing the notion of *viability*. A new predicate, *testViability*, was defined that called the viability predicate from the initial transformation model. However, *testViability* has as its parameters the *MeetingInitiator* as opposed to a general Actor. The results of running *testViability* projected over only one actor command is shown in Fig. 4.

The final test run was that of *workability*. This was conducted via a predicate, *testWorkability* that calls the "imported" workability predicate albeit with *MeetingInitiator* as its parameter as opposed to a general Actor. The resulting model from testing this predicate is shown in Fig. 5.

The three test models generated above are significant because they indicate the consistency of both the transformation model and the case study model. In order to utilise the predicates from the transformation model, a model under construction only needs to import the transformation model and call them using local parameters.

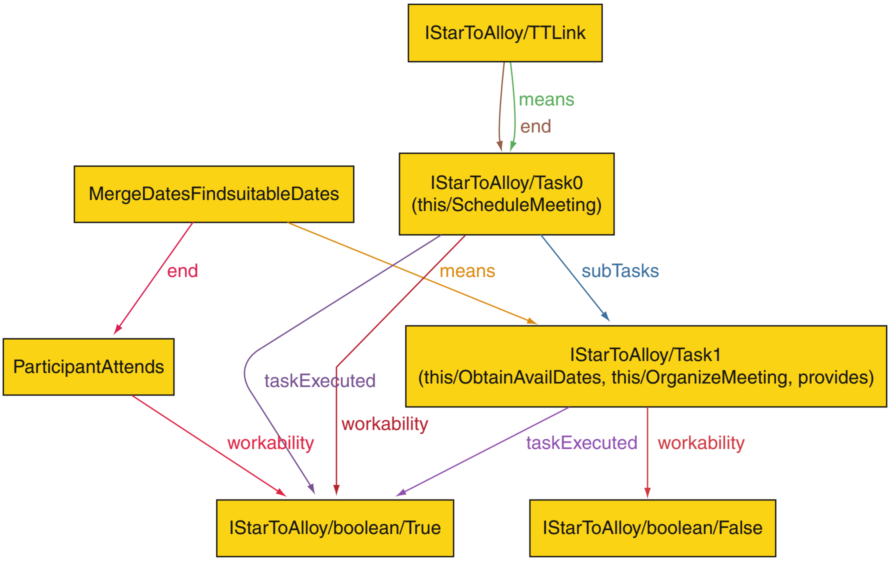


Fig. 4 Example model from running the viability predicate on the case study model

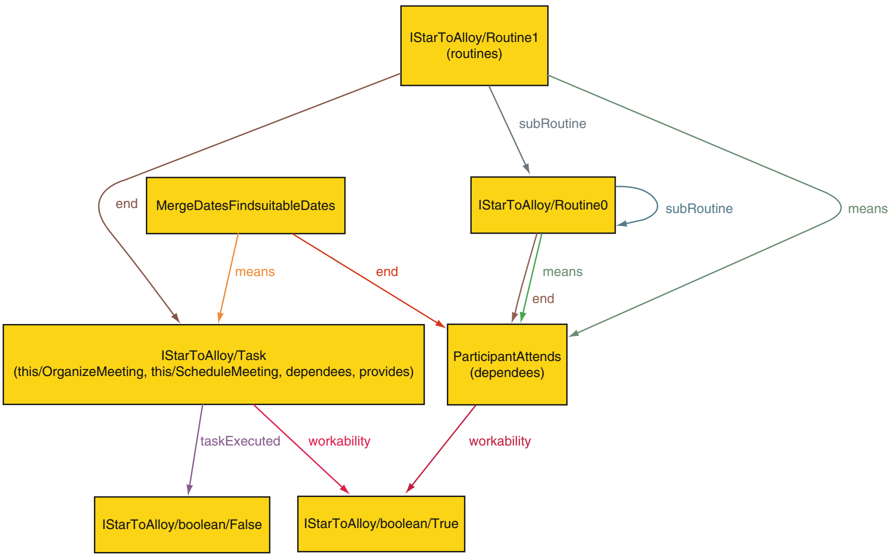


Fig. 5 Example model from running the workability predicate on the case study model

4 Conclusions and Future Work

Anstasakis et al. (2007) has done interesting work regarding UML analysis using Alloy. However, this approach doesn't consider the early stages of software development and in particular the early and late requirements engineering phases. Other work in this area includes that of Mostefaoui (2007) with regards to the verification of aspect UML models using Alloy. Alloy has also been used to expose hidden design flaws in a radiation therapy machine by Denis et al. (2004) as cited by Anstasakis et al. (2007). These approaches have different scope as compared to the work presented in this paper.

This discussion demonstrates the power of the Alloy model as a formal modelling notation that has tool support for automatic analysis. The *i** framework, on the other hand, provides a powerful technique for modelling and reasoning about systems and their environments. In this paper, an Alloy model has been developed that represents the *i** framework metamodel and was tested with the help of a case study. The results indicate the potential of this transformation model as a tool to assess software systems models before actual development. This will lead to software systems that better satisfy the users' needs as opposed to simply meeting the specified requirements.

Future work in the area of *i** and Alloy synergy could include further refinement on the concept of workability, which potentially would involve extending the *i** framework. Another area of interest would be the modelling of temporal elements of the processes (routines) of actors and in particular how they relate to each other i.e. whether they are parallel or sequential.

References

Anastasakis K et al (2007) UML2Alloy: a challenging model transformation. In: Model Driven Engineering Languages and Systems

Dennis G et al (2004) Automating commutativity analysis at the design level. In: Proceedings of the 2004 ACM SIGSOFT International Symposium of Software Testing and Analysis, New York, USA

Jackson D (2006) Software abstractions: language, logic, and analysis. The MIT Press, Cambridge

Krishna A et al (2009) Consistency preserving co-evolution of formal specifications and agent-oriented conceptual models. Inf Softw Technol 51(2): 478–496

Mostefaoui F (2007) Verification of aspect-UML models using alloy. In: Proceedings of the 10th international workshop on aspect-oriented modeling—AOM 07

Yu ESK (1995) Modeling strategic relationships for process reengineering. Ph.D. thesis. Department of Computer Science, University of Toronto

Yu ESK (1997) Towards modelling and reasoning support for early-phase requirements engineering. In: Proceedings of ISRE 97—3rd IEEE international symposium on requirements engineering

Planning and Scheduling of Business Processes in Run-Time: A Repair Planning Example

Irene Barba and Carmelo Del Valle

Abstract Over the last decade, the efficient and flexible management of business processes has become one of the most critical success aspects. Furthermore, there exists a growing interest in the application of Artificial Intelligence Planning and Scheduling techniques to automate the production and execution of models of organization. However, from our point of view, several connections between both disciplines remains to be exploited. The current work presents a proposal for modelling and enacting business processes that involve the selection and order of the activities to be executed (planning), besides the resource allocation (scheduling), considering the optimization of several functions and the reach of some objectives. The main novelty is that all decisions (even the activities selection) are taken in run-time considering the actual parameters of the execution, so the business process is managed in an efficient and flexible way. As an example, a complex and representative problem, the repair planning problem, is managed through the proposed approach.

1 Introduction

In the last years, the effective management of business processes (BP) in organizations became more important, since they need to adapt to the new commercial conditions, as well as to respond to competitive pressures, considering the business environment and the evaluation of their information systems. BP Management (BPM) can be seen as supporting BP using methods, techniques, and software to design, enact, control and analyze operational processes involving humans, organizations, applications, and other sources of information [2]. Similarly, Workflow Management Systems [1, 11] consist of methods and technologies for managing the flow of work in organizations. In a related way, BPM Systems (BPMS) are software tools that support the management of the BP. In some cases, they use temporal information

I. Barba (✉)
Universidad de Sevilla, Sevilla, Spain
e-mail: irenebr@us.es

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_6, © Springer Science+Business Media, LLC 2011

and ignore, in some ways, the required resources, considering them unlimited. This may not be adequate in different situations, for example when limited resources can be required by different activities at overlapped periods of time.

The area of Scheduling [8] includes problems in which it is necessary to determine an execution plan for a set of activities related by temporal constraints. Moreover, the execution of each activity requires the use of resources so they may compete for limited resources. In general, the objective is to find a feasible plan so that temporal and resource constraints are satisfied, usually optimizing objective functions related to temporal measures. In a wider perspective, in Artificial Intelligence (AI) planning [12], the activities to be executed are not established a priori, so it is necessary to select and to order them from a set of alternatives. In most cases, the specification of planning problems includes the initial state of the world, the goal (a predicate representing to a set of possible final states) that must be reached, and a set of operators (actions) that can be applied to one state to reach another one, allowing the evolution of the system. In order to solve this kind of problems, it is necessary to select a suitable set of actions that must be executed in a correct order, in some cases allowing parallelism. Furthermore, in several planning problems, the optimization of some objectives functions is pursued.

Currently, there is an increasing interest in integrating the application of AI Planning and Scheduling (P&S) techniques, some of them are maintenance and repair planning [10], where there may be a cascading set of choices for actions, facilities, tools or personnel, which affect different features of the plan, such as duration or cost [24]. Also, connection and disconnection planning involve the identification, selection and sequencing of operations, which can be specified by their effects on the components. In other context, disconnection planning has been object of different studies, such as maintenance or repair purposes [17]. Many problems can involve multiple conflicting objectives at the same time (multi-objective optimization [9]).

The And/Or graphs [15] (Sect. 3) can be used as a basis of representing most problems that involve both P&S, including the repair planning problem (Sect. 3), that is studied in the current work. A similar representation for this kind of problems can be found in [6], that proposes an extension of temporal networks by parallel and alternative branching to represent some kinds of P&S problems. Also, in [3] it is proposed the concept of blocks that can classify BP flows into several patterns: iterative block, serial block and parallel block including *AND* and *OR* structures.

BPM and AI P&S are two disciplines with many parallels, but which have largely been pursued by disjoint communities. BPMS are responsible for goals specification, design, implementation, enactment, monitoring and evaluation of BP [18], so they must deal with the anticipated P&S of processes. In general, automated P&S has not been integrated in business environments due to, between others, the lack of friendly modelling tools. There are several points where AI P&S tools can be effectively applied to BPMS, as it is explained in Sect. 2. In the last years, there exists a growing interest in the application of AI P&S techniques to automate the production and execution of models of organization, as shown by the existence of a Technical Coordination Unit of the European research network on P&S, PLANET [20], and

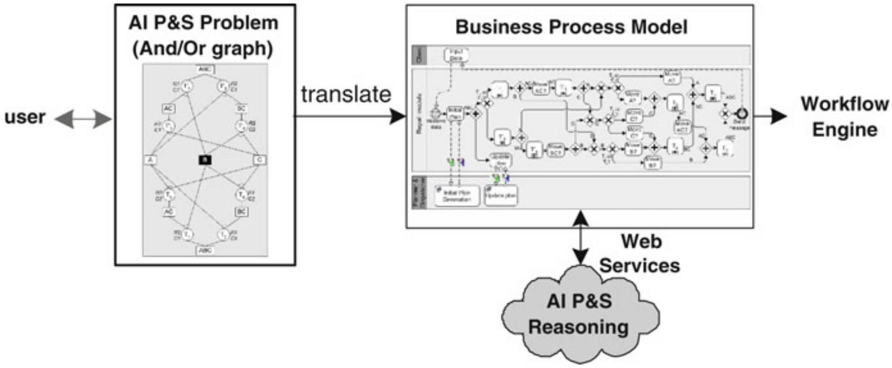


Fig. 1 AI P&S techniques for optimal business process enactment

several related research projects (SWIM system [7], the MILOS project [19], the TBPM project [16], etc). However, from our point of view, several connections between both disciplines remains to be exploited.

This work is based on modelling and enacting a problem involving both AI planning (selection and order of the activities) and scheduling (temporal resources allocation) through BP tools, using AI techniques in run-time for the optimization of several objective functions regarding the resulting execution plan, specifically duration and cost. The repair planning problem has been selected for being a complex and representative problem involving P&S. The most important contribution of this paper is the automatic management of the enactment of BP representing complex P&S problems, selecting and ordering the activities execution and allocating the resources in run-time, in order to avoid the drawbacks of taking these decisions during the design phase. Figure 1 shows a graphic representation of this work: first, the user introduces the information about the P&S problem, in this case, the repair problem. This information can be modelled through an And/Or graph or a BP model language, where all the execution alternatives are included, also containing the estimated values of several parameters, such as activity durations, resource availability, etc. This graph is translated to a BP model, that still represents a P&S problem since the selection and the order of the activities, together with the temporal resources allocation, are carried out in the enactment phase. The resulting BP model includes a pool composed by web services based on AI P&S techniques, that drives the execution considering the optimization functions and the actual values of the parameters. From this specification, the workflow engine can enact the plan in an optimized way.

In the BP enactment, the P&S decisions are taken considering the actual execution values instead of the estimated ones, optimizing the objective functions, so a flexible and dynamic BP is got. On the other hand, the decision-making in run-time provides the problem with a high spatial and temporal complexity. It is necessary to find a good balance between flexibility–optimality and complexity of the BP.

The main contributions of this paper can be summarized as:

- Optimal execution of BP that, in run-time, require: first, the selection and the order of the activities to be executed, due to the existence of several possible alternatives (planning) expressed through exclusive data-based gateways; and secondly, the resources allocation involving temporal reasoning due to the use of shared resources (scheduling), both considering the optimization of functions.
- Optimal execution of AI P&S problems that can be represented by And/Or graphs through BP tools, considering both P&S of the activities, and taking into account the optimization of objective functions.
- Translation from AI P&S problems that can be modelled through And/Or graphs, to BP models, focused on a complex and representative P&S problem: the repair planning problem.

In the following, Sect. 2 describes the BPM life cycle, Sect. 3 details the considered repair planning problem, Sect. 4 states the translation from the And/Or graph to a BP model, Sect. 5 summarizes the more related work and, finally, Sect. 6 presents some conclusions and future work.

2 AI Planning and Scheduling for BPM

The integration of AI P&S techniques with Workflow Management Systems can improve overall system functionality and can help to automate the definition of BP. As follows, the different stages of a typical BPM Life Cycle [2] are briefly presented:

- Process design: This stage involves designing, modelling, optimizing and simulating the organization processes. It is basically a human activity.
- System configuration: In this stage a schedule must be generated, taking into account the information given by the customer, such as the target end date, dependencies between activities or activity durations. The resources are assigned to activities for the appropriate time slots (scheduling), considering the finite capacity and/or non-sharable resources and, generally, taking into account the optimization of objective functions, such as process duration or cost.
- Process enactment: In this phase, the plan is carried out so that the activities have to be coordinated to ensure correct sequencing. At the same time, resources will be involved in enacting multiple processes and instances of the same process. When the same resource is required by several activities at the same time, generally, rules for prioritizing activities must act on the conflict.
- Diagnosis: As execution proceeds, the enactment information must be analyzed due to the possible appearance of unexpected incidents, such us unavailability of resources, actual activity durations different from expected ones, fails, etc. Minor incidents may require updating of the plan. More significant differences may require great changes in the plan, even a re-planning.

This work is based on automating the resource allocation and the selection and order of the activities, in both system configuration and process enactment phases, considering AI P&S techniques for optimizing several objective functions.

3 The Repair Planning Problem

Maintenance and repair planning is becoming a more important issue in many systems, according to the increasing problem of their complexity, the limitations of the technology used to maintain it and the problem of its ageing. In some studies, repair activities are integrated into the diagnosis process [10]. In order to repair some (previously diagnosed) faulty components, a sequence of disconnection activities must be executed to get it. After that, a repair action would substitute or repair the component, and then a set of connection activities must reconnect the system.

And/Or graphs have been used for the planning of assembly and repair processes [15]. This graph has been adapted for the representation of the proposed repair problem due to the existing similarities between both problems. The And/Or graph allows to represent the set of all feasible connection/disconnection plans in a natural way, depicting the system structure and including the alternative activities and the precedence constraints between them (Fig. 2a). It represents the disconnection (top part) and connection (bottom part) activities that can be selected in order to repair a faulty component (D in this case), in a system made of five components (*ABCDE*). In this representation two kinds of nodes can be distinguished:

- Or nodes: Correspond to subsystems, i.e., *AC* in Fig. 2. The top node corresponds to the complete system, and the leaf ones correspond to the components.
- And nodes: Correspond to the connection activities joining the subsystems of its two Or nodes below it producing the subsystem corresponding to the Or node above it, i.e., T_3; and the disconnection activities, that decomposes the subsystem above it to obtain the two subsystems below it, i.e., $T_1{}'$.

For the same Or node, there can be several And nodes (activities) below it, representing different alternatives to connect/disconnect the corresponding subsystem. In these graphs, each disconnection plan is associated to a tree, that is an And/Or path starting at the root node and ending at the nodes representing components (also, each connection plan is associated to a tree in a similar way). An important advantage is that the And/Or graph shows the activities that can be executed in parallel (Fig. 2). Furthermore, both precedence constraints and those related to the selection of activities, can be easily obtained from this representation.

A feasible repair plan can be seen as a (minimum) set of activities that starts with the disconnection of the complete system, repairs the faulty component, and finishes with the connection of the complete system. In this work, two kinds of activities are considered, each one presenting an associated duration and cost:

- Connection/disconnection activities: are executed on an established resource with a particular configuration to obtain the different subsystems.

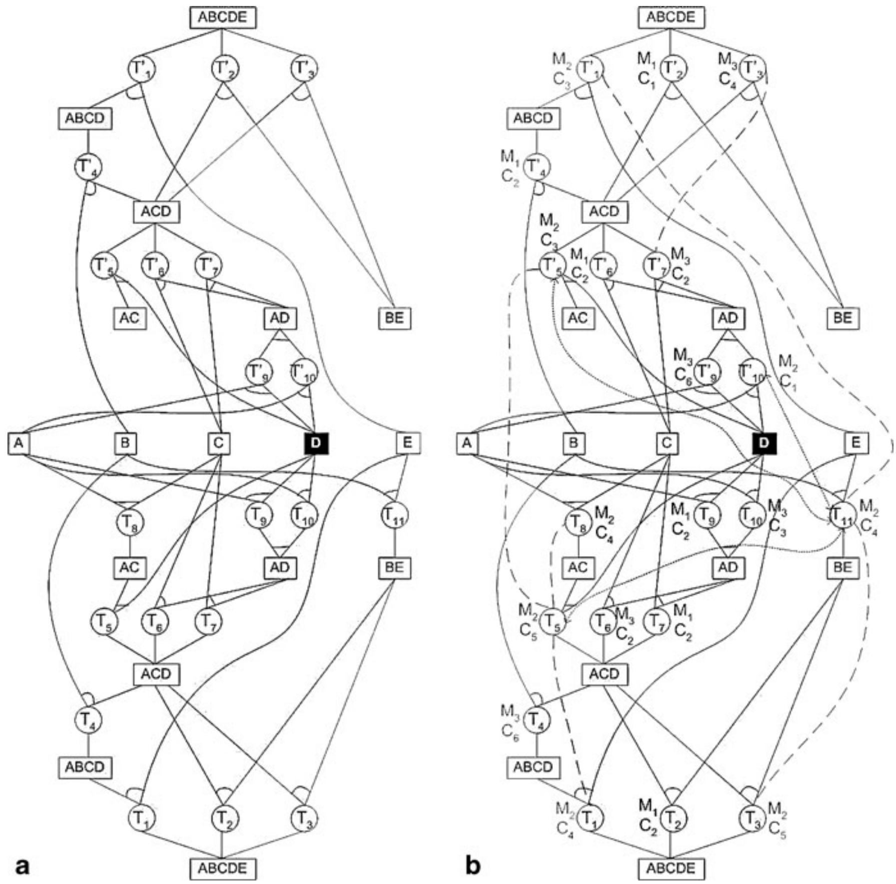


Fig. 2 Example of And/Or graph and an extension including resource constraints. **a** Original And/Or graph. **b** Extended And/Or graph

- Auxiliary activities: due to the use of shared and different resources. Two kinds of operations are considered: *set-up operations,* that change the configuration of a resource when two successive activities with different configuration use that resource; and *transportation operations,* that transport the subsystems between locations when the location where the subsystem is obtained after a connection/disconnection activity is different from the location where is required before a connection/disconnection activity.

In order to manage the shared resources with different configurations, the original And/Or graph has been extended, so that the new representation includes all the constraints involved in the problem, adding new types of links between the And nodes (Fig. 2b). The new links represent non-precedence constraints: due to the use of shared resources (i.e., T_5 and T_{11}) and due to the change of configurations in the resources (i.e., T_5 and T_1). The reasoning with resources, specifically temporal resources allocation, lead to NP completeness due to disjunctive constraints.

It is important to emphasize that the And/Or graph represents the system structure through components relations, that remains permanent despite of different required resources or faulty components. Consequently, for the same graph there can be several repair problems varying the faulty components or the required resources.

The And/Or graph can be used as a base representation of most AI P&S problems, since it allows to represent important P&S aspects, such as alternative activities or precedence relations. In general, this representation can be easily extended in order to include other aspects, i.e., shared resources in the repair planning problem.

4 The Repair Planning Problem as a Business Process

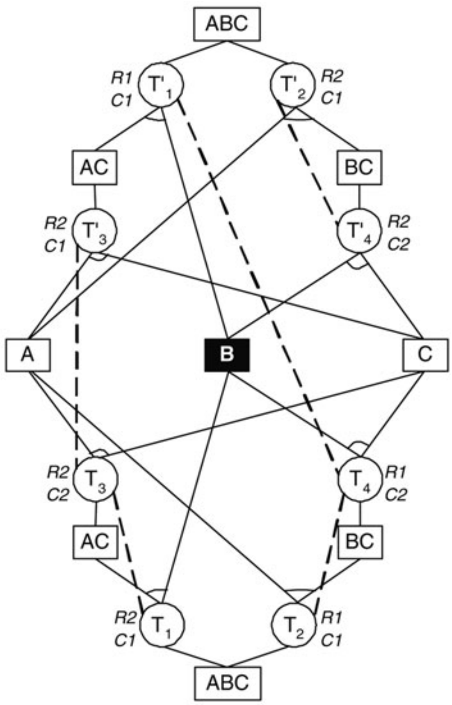
Once the problem in modelled as an And/Or graph, it can be translated to a BP language model, as instance BPMN [26], in order to be deployed and executed. It is important to emphasize that the And/Or graph contains all the possible alternative activities to be selected, and during the enactment phase, AI P&S methods are used to select and to order the suitable activities automatically, in order to obtain an optimal execution plan. As stated before, the And/Or graph represents the system structure through components relations, so different required resources and faulty components leads to different problems.

The BP model for the repair planning problem (Fig. 4) contains three pools:

1. *Client:* Abstract pool that acts as intermediary between the user and the BP. The user must specify the faulty components, that starts the BP enactment. At the end of the enactment, a message containing the selected activities together with their temporal allocation, is sent to the client.
2. *Repair module:* Executable pool that contains all the activities of the problem, from the And/Or graph specification.
3. *Planner and dispatcher:* Abstract pool that contains the web services. These web services are based on AI P&S techniques and are responsible for the optimal P&S of the activities. Each time that the repair plan execution *(Repair Module)* reaches OR branches, the *Planner and Dispatcher* pool decides the way that must be followed (planning). Furthermore, for the tasks requiring the same resources to be executed, this pool establishes the execution order (scheduling).

During the execution of the repair plan, the *Repair Module* and the *Planner and Dispatcher* pool are interchanging information. When the *Repair Module* reaches an exclusive data-based gateway, the decision to take is established by the information previously received from the *Planner and Dispatcher* pool. Different AI P&S techniques can be used in order to obtain an optimal execution plan. In previous works, we have developed proposals for solving the optimal P&S for the proposed repair planning problem, taking into account the minimization of both duration and cost. Specifically, in [4] it is explained a constraint based proposal [23] and in [5] it is shown a PDDL specification for generic planners [12], both considering multimode

Fig. 3 Example of extended And/Or graph for a system made of three components when faulty component is B



activities and multi-objective optimization (duration and cost). In both, several experimental results are shown, demonstrating that these proposals are successful at solving this kind of problems. These tools have been used for developing the web services related to P&S in the current work.

As follows, the BP model generation from the And/Or graph is explained. As an example, the BPMN corresponding to the repair problem of Fig. 3 can be seen in Fig. 4. For the sake of clarity, in Fig. 4a the activities corresponding to connection/disconnection tasks appear collapsed, while in Fig. 4b its expanded representation is shown. Regarding to the proposed BP model, first of all, a message containing the faulty components is sent from the *Client* pool to the *Repair Module*. After that, an initial plan is generated trying to obtain a good start point (optimizing the objective functions). This initial plan, in most cases, will be iteratively improved during the BP enactment, as explained later. After the initial plan generation, two branches are executed in parallel in the *Repair Module*:

- Optimization process (Branch 1 in Fig. 4a): The initial plan can be non-optimal. Also, the actual parameters can be different from the estimated ones. Considering both reasons, it is proposed a loop updating activity that is continuously trying to improve the current solution, taking the actual values of the parameters. In order to do this, a web service based on AI techniques *(Update plan)* is invoked. Each time a better solution is found, the current plan is updated and these information

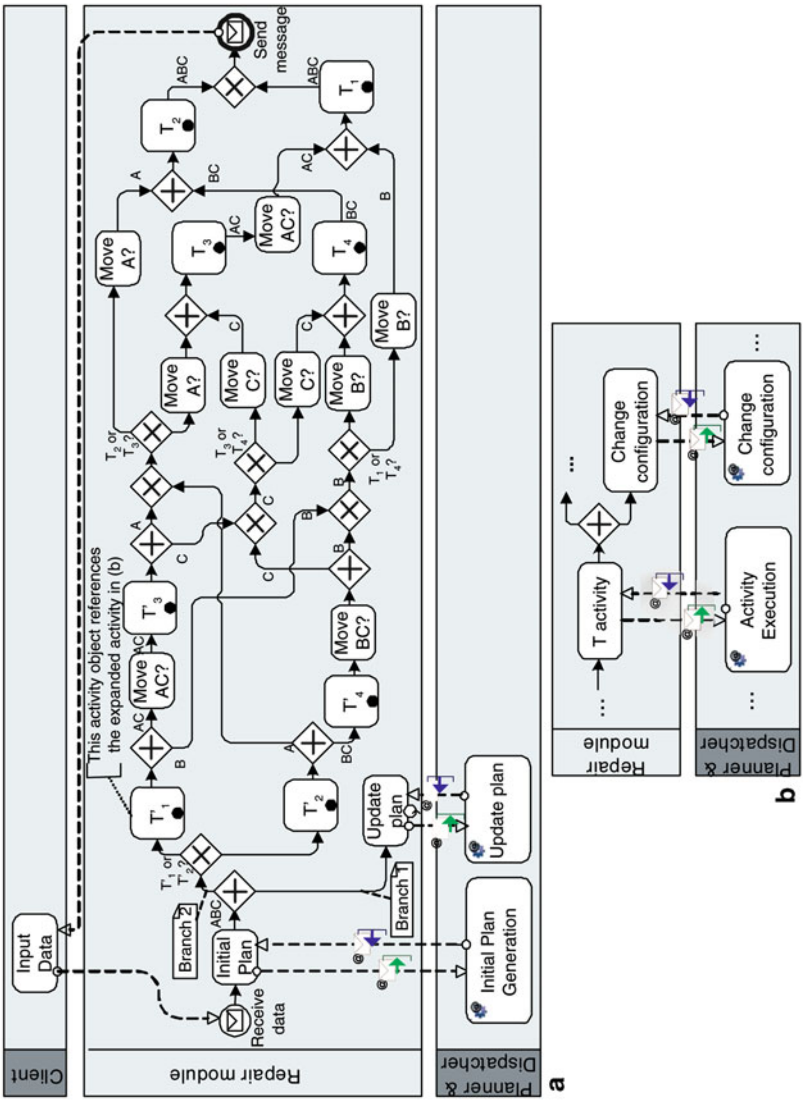


Fig. 4 BPMN diagram of the repair problem of Fig. 3. **a** BPMN diagram with collapsed activities. **b** BPMN diagram for the expanded activity

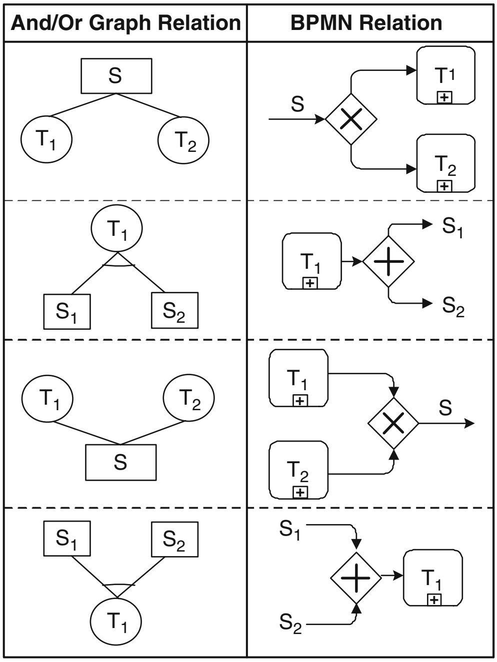


Fig. 5 Translation from And/Or graph to BPMN relations

is used to decide the way to follow in the exclusive data-based gateways. The loop finishes when the repaired complete system is successfully obtained.

- Repair plan (Branch 2 in Fig. 4a): the activities are executed considering the P&S established by the *Planner and Dispatcher,* as follows:
  - Each relation that exists in the And/Or graph between several components is translated to another relation in the BPMN model (Fig. 5).
  - The location where the subsystem is obtained after a connection/disconnection activity can be different from the one where it is required before a connection/disconnection activity, so it is necessary to consider the activity *Move subsystem?* between two successive activities that act on the same subsystem. The *Move?* activity has to be executed only when the successive activities take place in different locations (in the current work, different resources are considered different locations). For the *Move?* execution, it is not necessary to call web services of *Planner and Dispatcher* pool since the information about the location where the subsystem was obtained can be consulted in the plan execution, and the information about the location where it is required depends on the selected branch. If a subsystem will be moved or not during the enactment phase, it depends on the plan execution because of different activities can obtain a subsystem, and it can be required also for different ones.
  - Regarding to Fig. 4b, *T Activity* is the actual execution of the connection/disconnection task. A web service must manage it, since it must establish

the start time of the task, its duration, etc. Taking into account the task execution information, the actual values are stored, and used for the plan update and the rest of the BP enactment. The *Change of configuration* activity, provided that it is necessary, must be done in parallel with the treatment of the subsystems obtained after the activity execution (Fig. 4b). The *Change of configuration* activity is also controlled by the *Planner and Dispatcher* module since its execution causes the resource lock, that influences the BP execution. Also, if this activity is executed or not, and when it is executed, depend on several factors that must be analyzed by the *Planner and Dispatcher* pool.

Finally, when the repair plan has been successfully executed, a message with the plan details is sent to the client.

5 Related Work

Several research groups have integrated AI techniques with BPM systems. As follows, some of the similar works are briefly described, explaining the main differences between them and the current approach.

Some related works integrate P&S tools in BPM systems for the modelling phase, such as [13], where a user specifies the information related to the BP through an execution/interchange standard language, XPDL, that is translated to HTN-PDDL language (an extension of PDDL) in order to obtain suitable timeframe for each activity and resource allocation. In a similar way, in [22], planning tools are used to generate BP models, taking into account the knowledge introduced through BP Reengineering languages. This knowledge is translated into predicate logic terms in order to be treated by a planner that integrates P&S, and obtains an updated BP model. Both works generate the BP model during the build-time, without taking into account the actual values of the parameters obtained in run-time. Consequently, the initial optimization plan can be obsolete due to the non-updating aspect. On a different way, the current proposal, in order to update the plan in run-time, maintains all the possible alternatives in the BP model, taking into account the actual values of the parameters for updating the P&S of the activities.

On the other hand, most of the related works integrate scheduling tools in BPM systems for the enactment phase, in order to take dispatching decisions as to which activity should be executed using a resource when it becomes free (dynamic scheduling). As follows, a representative set of them are briefly summarized. One of the first works, [27], developed a framework for temporal workflow management, including turnaround time predication, time allocation, and task prioritization. In a related way, [25] proposes a schema for maximizing the number of workflow instances satisfying a predetermined deadline, based on a method to find out the critical activities. Also, [14] proposes a set of process execution rules based on individual worklists, and it develops algorithms for the task assignment in order to maximize the overall process efficiency, taking into account the resources capacities. Recently, [21] presents a Theory of Constraints (TOC)-based method for improving the efficiency of BP. There are several differences between these works and the current approach. First, in the current proposal the selection of the activities to be executed (planning) is developed in run-time, besides the resource allocation, while the previously summarized works only consider the prioritization of the tasks using the same resource (scheduling). On the other hand, the current approach considers several objective functions, while the other works only focus on the temporal aspects of the processes for allocating the resources.

To the best of our knowledge, there is not any proposal for selecting the adequate activities in run-time, in order to optimize some functions in BP environments and considering reaching several objectives (planning).

6 Conclusions and Future Work

The current work is based on modelling and enacting a representative and complex problem through BP tools, the repair planning problem, involving both planning and scheduling aspects. The enacting of the BP is managed by AI techniques in run-time, so that the activities to be executed are selected and ordered, and the resources are temporarily allocated. The main novelty of the work is the planning of the activities in run-time, taking into account the optimization of several functions and the reach of some objectives. On the other hand, the proposed architecture and AI techniques can be used for the optimal execution of different BP (expressed through a model language) that, in run-time, require: first, the selection and the order of the activities to be executed, due to the existence of several possible alternatives (planning), and secondly, the temporal resources allocation due to the use of shared resources (scheduling), both taking into account the optimization of objective functions.

As future work, it is intended to analyze another kinds of BP problems involving planning and scheduling aspects. Also, a bidirectional translation between planning languages, as PDDL, and BP model or execution languages, is intended to be explored. Furthermore, another objective functions can be considered.

Acknowledgements This work has been partially funded by the Consejeria de Innovacion, Ciencia y Empresa of Junta de Andalucia (P08-TIC-04095) and by the Spanish Ministerio de Ciencia e Innovacion (TIN2009-13714) and the European Regional Development Fund (ERDF/FEDER).

References

1. van der Aalst WMP, van Hee KM (2002) Workflow management: models, methods, and systems. MIT Press, Cambridge
2. van der Aalst WMP, ter Hofstede AHM, Weske M (2003) Business process management: a survey. In: Business process management: international conference, BPM 2003, Eindhoven, The Netherlands, 26–27 June 2003 (Proceedings 1–12)
3. Bae J, Bae H, Kang S, Kim Y (2004) Automatic control of workflow processes using ECA rules. IEEE Tran Knowl Data Eng 16(8):1010–1023
4. Barba I, Del Valle C, Borrego D (2009) A constraint-based model for multi-objective repair planning. In: 14th IEEE international conference on emerging techonologies and factory automation

5. Barba I, Del Valle C, Borrego D (2009) PDDL specification for multi-objective repair planning. In: Caepia 2009 workshop on planning, scheduling and constraint satisfaction, pp 21–33
6. Bartak R, Cepek O (2007) Temporal networks with alternatives: complexity and model. In: Proceedings of the twentieth international florida artificial intelligence research society conference, FLAIRS 2007, pp 641–646
7. Berry PM, Drabble B (2000) SWIM: an AI-based system for organizational management. In: The 2nd NASA international work shop on planning and scheduling for space, NASA, Washington, DC
8. Brucker P, Knust S (2006) Complex scheduling (GOR-Publications). Springer, New York
9. Coello CA (2006) Evolutionary multi-objective optimization: a historical view of the field. IEEE Comput Intell Magaz 1(1):28–36
10. Friedrich G, Gottlob G, Nejdl W (1994) Formalizing the repair process—extended report. Ann Math Artif Intell 11:187–201
11. Georgakopoulos D, Hornick M, Sheth A (1995) An overview of workflow management: from process modeling to workflow automation infrastructure. In: Distributed and parallel databases, pp 119–153
12. Ghallab M, Nau D, Traverso P (2004) Automated planning: theory and practice. Morgan Kaufmann, Amsterdam
13. Gonzalez-Ferrer A et al (2008) Towards the use of XPDL as planning and scheduling modeling tool: the workflow patterns approach. In: 5290 LNAI, pp 52–61
14. Ha B, Bae J, Park Y, Kang S (2006) Development of process execution rules for workload balancing on agents. Data Knowl Eng 56(1):64–84
15. Homem de Mello LS, Sanderson AC (1988) Planning repair sequences using the And/Or graph representation of assembly plans. IEEE international conference on robotics and automation, pp 1861–1862
16. Stader J, Moore J, Chung P, McBriar I, Ravinranathan M, Macintosh A (2000) Applying intelligent workflow management in the chemicals industries. In: Fisher L (ed) The workflow handbook 2001, Published in association with the Workflow Management Coalition (WfMC), Oct 2000, pp 161–181, ISBN 0-9703509-0-2
17. Li W, Zhang C (1995) Design for disassembly analysis for environmentally conscious design and manufacturing. Proc 1995 ASME Int Mech Eng Cong 2:969–976
18. Muehlen MZ, Ting-Yi Ho D (2005) Risk management in the BPM lifecycle. In: Business process management workshops, pp 454–466
19. Münch J, Goldmann S, Holz H (2000) Distributed process planning support with MILOS. Int J Softw Eng Knowl Eng 10(4):511–525
20. Network home page: European Network of Excellence in AI planning. http://planet.dfki.de/index.html. Accessed 27 June 2011
21. Rhee S, Cho NW, Bae H (2008) Increasing the efficiency of business processes using a theory of constraints. Inf Syst Front 12(4), 1–13. doi:10.1007/s10796-008-9145-9
22. R-Moreno MD, Borrajo D, Cesta A, Oddi A (2007) Integrating planning and scheduling in workflow domains. Expert Syst Appl 33(2):389–406
23. Rossi F, Van Beek P, Walsh T (2006) Handbook of constraint programming. Elsevier, Amsterdam
24. Smith D, Frank J, Jonsson A (2000) Bridging the gap between planning and scheduling. Knowl Eng Rev 15(1):47–83
25. Son JH, Kim MH (2001) Improving the performance of timeconstrained workflow processing. J Syst Softw 58(3):211–219. doi:10.1016/S0164-1212(01)00039-5
26. White SA et al (2004) Business process modeling notation (BPMN), working draft, Version 1.0. BPMI.org
27. Zhao JL, Stohr EA (1999) Temporal workflow management in a claim handling system. SIGSOFT: Softw Eng Notes 24(2):187–195. doi:10.1145/295666.295687

A Study of First Click Behaviour and User Interaction on the Google SERP

Chris Barry and Mark Lardner

Abstract Firms use Search Engine Marketing (SEM) to drive users to their Website. Some are prepared to pay for placement; others use Search Engine Optimization (SEO) hoping their result percolates up the organic SERP. Despite extensive SEM efforts, firms can only speculate over the first critical interaction between the first SERP and a user's first click. This study sheds some light on users' first click behaviour on Google and the early interaction thereafter. The research reveals that users evaluate the SERP from the top downwards, deciding instantly whether to click into each link, while first clicks are predominantly at the top of the SERP, especially towards organic links. For certain queries top sponsored links received almost as many clicks as organic links despite what users profess. Recommendations to firms include advice that strategies should be primarily SEO focused and that paid search campaigns should maintain a position in the top sponsored links section of the Google SERP.

1 Introduction

Firms use search engine marketing (SEM) to drive users to their Website. There are two main strategies: paid search and search engine optimization (SEO). two strategies are not mutually exclusive and various arguments can be made for pursuing each strategy (Barry and Charleton 2008). Whichever SEM strategy is chosen, competition for a high search engine results page (SERP) ranking is intense. Firms can make critical decisions about SEM strategy if they have insight s into early user interaction. This study seeks to provide such insights on the SERP of the predominant search engine (Lewandowski 2008), Google.

C. Barry (✉)
National University of Ireland, Galway, Ireland
e-mail: chris.barry@nuigalway.ie

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_7, © Springer Science+Business Media, LLC 2011

2 The Nature of Search Online

Searching online refers to the input of a query (a list of one or more keywords) into the text box of a search engine followed by a list of results returned on the SERP (O'Brien and Keane 2006). The abundance of content online makes firms ever more reliant on the search engine optimally indexing and ranking their Web pages so they are found in the earliest part of a user's interaction. Search engines determine these ranks through their own typically undisclosed ranking algorithms (Bar-Ilan 2007). Failing to find what they want after only the first click could create negative perceptions of the search engine and the chosen Website and possible reputational damage. Furthermore, search engine algorithms are increasingly more impressed when users actually click on a link for any given query (Ciaramita et al. 2008). It is also becoming clearer how little time users spend making judgments on the relevancy of links and how easily distracted or impatient they can be (Browne et al. 2007; Szetela 2008).

A SERP consists of two main areas: the visible area and the scrolling area (Hochstotter and Lewandowski 2009). Information that is immediately seen on a screen is the visible area of the SERP while the scrolling area must be scrolled to. In Hochstotter and Lewandowski's study the SERP links that occur "below the fold" (i.e. links only visible when the user scrolls) are very rarely clicked upon. This increases the pressure on firms to ensure their site appears above the fold. An important characteristic of user behaviour is that very few users actually move on to the second SERP after performing a search query (Bar-Ilan et al. 2006; Richardson et al. 2007). Furthermore, a recent report found that 68% of all users concentrate their clicks on the first SERP (iProspect 2008). This makes competition for a high ranking very intense and vulnerable to ranking fluctuation at all times.

The results that are returned by a search engine with any given query are divided into two sections; organic listings and sponsored listings (Jansen and Resnick 2005; Jansen and Molina 2006). Organic links are links that the search engine's algorithm deems most relevant to the query, whilst sponsored links are links that appear as a result of a party purchasing the right to be displayed on the SERP for a particular keyword (Jansen and Resnick 2005).

Barry and Charleton (2008) report that SEM strategies are the predominant methods firms use to promote their visibility and that the primary objective of an SEM strategy is to ensure that one's site is displayed at or near the top of the first SERP. Paid search allows firms or organisations to place bids on certain keywords and hence have their advertisements displayed along side or above the organic listings as sponsored listings (Bar-Ilan 2007). The higher ranked sponsored listings typically cost more. Payment may also be made for a click through to a Website. SEO involves altering a Web page so that it appears to be more relevant to the search engines. SEO aims to exploit search engine algorithms to attract users to specific Websites presented as organic results (Barry and Charleton 2008). SEO is carried out by altering the aspects of the Website (such as relevant keywords in the metadata of a page) that a search engine's "spider" is thought to explore in judging relevancy (Jones 2008). In a recent study of Irish small to medium enterprises (SMEs), it was found that the majority of SMEs show a preference for SEO over paid search (Barry and Charleton 2008).

One of the chief methods that Google uses to evaluate the full extent of the relevancy of a page is through a metric called "PageRank" which helps measure the popularity of a page (Cho and Roy 2004). The page popularity is determined by how many other Web pages link to a specific page on the Internet. A simplified example—Google will present a hyperlink at the top of its SERP if that hyperlink contains the relevant keywords in its metadata and also if that hyperlink is linked to by more external sites than other hyperlinks with similar keywords (Cho and Roy 2004). Google uses complex algorithms to rank the results organically on the SERP. The highly optimised pages (via SEO) are those more likely to appear on the first SERP (Jones 2008).

3 Evaluation of the SERP

SERPs are presented in some form of text based summaries and based on the information contained within these summaries users make relevance judgments based on what links best suit their information needs (Jansen and Resnick 2005; O'Brien and Keane 2006). The question of how users evaluate the relevance of the SERP can be answered through the investigation of users' first click behaviour. First click behaviour describes "the process of when a user (a) poses a query to a search engine to fulfill some information need, (b) evaluates the results list returned to that query and (c) then chooses one of these results as a link to follow" (O'Brien et al. 2006, p. 1).

Klockner, Wirschum and Jameson contend users either adopt a depth-first or a breadth-first approach (2004). A depth-fist approach refers to when users evaluate each entry in the list in turn, starting from the top, and deciding systematically whether to choose that link or to move onto the next consecutive link. A breadth-first approach is where a user looks ahead at a number of list entries and then revisits the most promising ones in order to open them. In an eye tracking study conducted by Klockner et al. (2004), 65% of participants utilised the depth-first approach with the remaining participants adopting a fully or sometimes partially breadth-first approach.

Keane et al. (2008) showed that users do have a tendency to choose results at the top rather than results down the bottom of a list, but this tendency will not be as prevalent when the relevancy of the top result is weakened. Joachims et al. (2005) contend that users' trust in the search engine mean links ranked first received many more clicks than the following link and also that the top two links received more attention than the rest of the SERP. Opinions are divided as to whether a click on a link is representative as a vote of relevance. Joachims et al. (2000) believe that a click on a hyperlink from a user on the SERP is representative of an endorsement of that page. On the other hand, Ciaramita et al. (2008) believe that clicks on particular links are not an indication of absolute relevance, but only indicates that the clicked item is perceived to be more relevant than other links that were ranked elsewhere but not actually clicked.

While users have been shown to examine both organic and sponsored listings, organic links are ultimately perceived as being more relevant (Jansen et al. 2007), even if sponsored links are reported to be just as relevant as organic links (Jansen and Resnick 2005). Laffey also shows that with Google specifically, users are more likely to click on organic links, rather than sponsored links (2007). Users have also expressed negative emotional reactions to sponsored search listings (Marable 2003). Most search engines including Google have a vested interest in hoping that sponsored search has a future since their business model depends significantly on it. Previous research also emphasises the importance of appearing high in the rankings for sponsored listings (Richardson et al. 2007).

Queries are often divided into transactional, informational and navigational queries. Informational queries are queries seeking images, songs, videos or documents while transactional ones are queries with a commercial motive (purchasing a specific product or service) and navigational-based queries are entered to take users to a specific URL (Jansen et al. 2005). Previous studies have shown that the majority of users use search engines to find out specific pieces of information or to ask questions (Rose and Levinson 2004; Zhang et al. 2009). Jansen et al. provided some useful insights into user perception of sponsored and organic links for transactional queries (2007): sponsored links are likely to be more relevant than organic links for e-commerce queries; top listed sponsored links are more relevant than side sponsored; and the rank of side-sponsored links are not correlated with the relevance of sponsored links. A recent study showed that 80% of Web search queries are predominantly informational with the remaining 20% of queries split between navigational based and transactional-based queries (Jansen et al. 2005).

4 Research Objectives and Methodology

Despite extensive SEM efforts, firms can only speculate about the first critical interaction between the return of the first SERP and a user's first click. Thus the main objective of the study is an investigation of how Web searchers perceive and interact with the Google SERP for informational and transactional queries from a first click behavioural perspective. Secondary objectives are: to explore the nature of user's link assessment strategy for both informational and transactional queries; to reveal how favourably users view organic versus sponsored links; and to identify where, within the Google SERP, are first clicks most concentrated.

In order to reveal a rich picture, both quantitative and qualitative research methods were used. Three research techniques were used: verbal protocols, interviews and Web based questionnaires. Verbal or "think aloud" protocols refer to the practice of collecting data by getting participants to think aloud as they partake in a set of specific tasks (Crutcher 1994). Verbal protocols were used to examine how users perceived and interacted with the Google SERP from a first click behaviour perspective, interviews to examine in more detail the issues and concerns arising from the verbal protocol sessions, and Web based questionnaires were administered in order to capture necessary participant demographics. Each participant, under controlled laboratory conditions, was asked to respond to the most common types of Google searches—one using an informational query and another using a transactional query (Jansen et al. 2005). These types of queries were used since research shows that they are the two most common types of queries (Zhang et al. 2009). Both tasks were very specific, achievable and easy to explain to a novice user. All verbalisations and on screen behaviour (e.g., clicks, scrolling, mouse movements) were recorded by an open source computer programme called "CamStudio". The combination of verbal protocols with a simultaneous recording of on-screen activities has shown to be a valuable observational method, previously used by Van Waes (1998). For the study, a convenience sample of twenty participants was used.

5 Research Findings

5.1 Link Assessment Strategy Analysis

A variant of the depth and breadth-first approaches of Klöckner et al. (2004) to analyse first click behaviour was used. This more sophisticated analysis identifies usage patterns that characterise each approach and combines verbalised thoughts to more fully understand user behaviour. These patterns are identified in Table 1 and represented in Fig. 1.

In the informational query sessions, the variant depth-first approach was significantly more evident with 17 out of the 20 participants characteristically using it, without any scrolling action before the first click. Each first click was on the first organic link and verbalisations made in tandem with the first clicks included: ". . . guess I should go for the first one" and "First up is Wikipedia which I usually use. I'm going to try that one first". There was little hesitation and behaviour was hurried.

Table 1 Usage patterns of link assessment approaches

Link Assessment Approach	
Variant Depth-First	**Variant Breadth-First**
Characterised by: An immediate first click without any partial/full scroll Sequential/non-sequential click-through pattern starting from top to bottom of SERP Verbalisations (e.g. "First thing I'll do is click the top one") Behaviour appears more impatient and rushed	**Characterised by:** First click being preceded by partial or full scroll through SERP (Then a retreat up the SERP) Participant placing cursor over one or more links momentarily before retreating up the SERP for first click Verbalisations (e.g. "I'll just look through the list first to see what else there is...") Behaviour appears more thorough and aware

😐 = One Participant	Informational (Query)	Transactional (Query)
Depth-First (Link Assessment)	😐😐😐😐😐😐 😐😐😐😐😐😐 😐😐😐😐😐	😐😐😐😐😐😐 😐😐😐😐😐😐 😐
Partial/Full Breadth-First (Link Assessment)	😐😐😐	😐😐😐😐😐😐 😐

Fig. 1 First click behaviour: incidence of depth-first and breadth-first link assessment strategies

The action appeared a somewhat automated response, entrusting the first result on the SERP with a confident assertion of relevancy. Depth-first was also more common for the transactional session with 13 participants utilising the approach. Depth-first approaches came in a mix of sequential and non-sequential click-through patterns on the SERP. Some participants evaluated the list sequentially, e.g. "... I'm going to go down to the next ones... there's Amazon (scroll), Play (scroll), there's a Dublin one so I'm going to go for that one". Other depth-first instances were more instantaneous "... Amazon is the first one I'm going to". While search sessions did take longer for participants to complete than for the informational query, the hurried depth-first approach was still more regular.

Overall, instances of breadth-first search were far fewer for both the informational and transactional sessions. Only three participants exhibited the breadth-first approach for the informational search. One participant performed a partial scroll of the SERP before returning to the top link, stating: "... Wikipedia is your best bet". The lack of a breadth-first strategy when informational searching online could be indicative of impatience, consistent with suggestions of Browne et al. (2007) and Szetela (2008) above. There were notably more breadth-first usage for the transactional query. Breadth-first search is exhibited through the following illustrative verbalisations: (a) "I'm scrolling down to see if there is any link that immediately catches my eye... "; (b) "I'm going to scroll down to see if there is anything else that looks good before I click into that. There's lots of reviews and stuff, not looking for that. I'm going to go for Amazon.com"; and (c) "OK, any of these I recognise? Don't know if they're good... hmmm... (Scrolls to bottom). See do I recognise any first before I move on... ". Another participant displayed signs of breadth-first search by musing "E-bay, Amazon (third top sponsored and first organic links respectively) kind of stick out to me anyway... sure I'll look at the first one... ".

These combined results show that most participants (75%) are not prepared to pour over the entire SERP in detail, instead preferring to adopt a more hurried depth-first approach whereby the user makes more spontaneous and rushed judgments when deciding where to click first. This finding also correlates with the study by Klöckner et al. (2004) who found that the depth-first approach was used 65% of the time, with the remaining users adopting a breadth-first search.

5.2 User Perceptions of Organic and Sponsored Links

The click data collected from the verbal protocol sessions produced some interesting results. As mentioned above all participants chose the top organic link as their first click for the informational query. In the absence of sponsored links, participants tend to gravitate towards the first organic link, irrespective of the link assessment strategy. Of all the first clicks carried out on the transactional session 11 (55%) of the first clicks were on organic links, confirming a disposition towards the use of organic links (Laffey 2007). In this case, the first organic link received the most clicks.

Anomalously, despite nine participants (45%) choosing a sponsored link as a first click, verbalisations and post-test feedback indicated that sponsored links are not deemed relevant. One participant suggested that the sponsored links are automatically ignored: "... your train of thought is to automatically go to the first link up here below the sponsored links". Other verbalisations also show an explicit disregard for sponsored links: "... they're sponsored links. I never go for them, so I'm going to go down to the next ones". Sponsored links were also being viewed as having "... catches in them." and not being "... your best bet for getting a deal", almost a hindrance. This view concurs with the finding of Jansen et al. (2007) that stated that there were no strong relevance correlations with sponsored links. Conceivably, the anomaly might be explained by some confusing the organic and sponsored areas. A number of participants did not seem to distinguish the top sponsored links section as being separate to the organic links with one participant referring to the first organic link as the "... fourth one on the list". Another participant expressed difficulty differentiating between organic and sponsored links.

In contrast, organic links are viewed far more favourably, described by one as the "... normal area...", another commenting "I'd prefer to use the non sponsored links because they're more impartial...". Consistent with other studies participants appeared to find organic links more relevant than sponsored links (Hotchkiss et al. 2004; Jansen and Resnick 2005; Jansen and Molina 2006; Jansen et al. 2005).

The actual search engine behaviour of users and the comments in the post-test interview, that somewhat contradict their previous action, could be explained by the concepts of espoused theory and theory-in-use (Argyris 1980). It may be the case that users overtly subscribe to some notion of the purity of organic search (espoused theory), but in the hurried world of online search, they are often prepared to compromise by utilising all available links (theory-in-use).

All first clicks were concentrated entirely on top sponsored links rather than right-hand sponsored links. This correlates with research from Jansen et al. (2007) that found no real relevance correlation with right-hand sponsored links. This finding is also backed up with a remark from one participant who noticed that one link in the right-hand sponsored section was a dating site as opposed to an e-commerce based site for DVDs. Right-hand sponsored links received far more negative attention than top sponsored links throughout the transactional session. They were described as "... just advertisements that I don't want to be wasting my time checking" and "... untrustworthy...". Another participant remarked dismissively that "... if they were any good they would be in the main search".

5.3 First Click Distribution on the SERP

For the informational query all participants' first clicks were carried out on the first organic link. This finding is consistent with previous research where it was found that links ranked first in the SERP, received many more clicks than the following link on the SERP (Joachims et al. 2005). The same research also found that users pay most attention to the first and second links on the SERP.

In respect of transactional queries, 95% of all first clicks occurred between the top sponsored link and the third organic link (within the visible area of the SERP), again consistent with Joachims et al. (2005). The pattern between these links is that the first top sponsored link received four more first clicks than the next link and the first organic link received five more first clicks than the following organic link. This emphasises the importance of ensuring that SEO secures a place in the "visible" area of the screen (Hochstotter and Lewandowski 2009).

Verbalisations from the interviews also reveal a tendency to gravitate towards the top links on the SERP. "I usually just go to the one on the top of the page. . . if it doesn't suit me I just go next one down". Some participants display a lack of patience when describing their online search habits: "If I don't get it, at a push, in the first five links, more often than not I just give up, I just go and type in something else", consistent with the notion that it only takes one second for a user to make a relevancy judgment (Szetela 2008). The apparent urge shown by participants here to act quickly on the SERP and then refine the query if no relevancy is found at the top demonstrates that users are becoming increasingly dissatisfied with the search engine, the longer the time they spend on it (Browne et al. 2007). Participants did seem to possess an explicit trust in the top half of the SERP: ". . . even if you're not sure what you're looking for, you assume that the first result will be relevant"; "The first one that comes up is usually the most relevant"; "generally I will click on the top few". Verbalisations like these suggest that SEO strategies must be focused on a high SERP ranking if they are to have any chance of winning the race for that first click. These findings and verbalisations also tie in with previous research on this topic, that conclude that users have a tendency to concentrate mainly on the top part of the SERP whilst paying significantly less attention to the bottom part (O'Brien and Keane 2006; Ciaramita et al. 2008; Keane et al. 2008).

6 Conclusions

In light of the above findings and from an SEM perspective, companies would be advised to make the achievement of a high-ranking position on the Google SERP the primary objective of their SEM strategy. The findings demonstrate that companies are at a distinct disadvantage if their link does not appear in the visible area of the Google SERP. This distribution towards the top is also noted in other studies (Richardson et al. 2007; Keane et al. 2008). If it is accepted that users place more trust in the top two links (Joachims et al. 2005), users who utilise a depth-first approach will execute that vital first click on the top part of the SERP. Therefore, SEM strategies should work under the assumption that users assess the SERP using the depth-first approach. SEM strategists should therefore take all steps necessary (on site optimization and paid search campaigns) to ensure that their site appears and stays in a high-ranking position on the Google SERP in order to increase traffic to their site. Exploiting PageRank is key to achieving this strategy. In this regard, Google could assist firms by offering more clarity on the reasons why a link is on top of the SERP.

SEO specialists should also ensure that users who click through to their link on the SERP are greeted with a landing page that meets the needs of the user and correlates sufficiently with the apparent relevancy of the SERP link. This is critical so that users do not feel compelled to navigate back to the SERP and click into a rival site (Todd 2006; Szetela 2008). Sites not ranked highly on the Google SERP that are effectively less "popular" need to design their SEO strategy that views the first Google SERP as a highly competitive market where the onus is on the lower ranked sites to justify and demonstrate to Google why their sites should be placed at a higher ranking. This can be achieved if an SEO-focused SEM strategy is designed to encompass as many of the reported factors Google use to determine relevancy with special attention given to the PageRank metric. To achieve this, a company needs skilled individuals to ethically implement continuous SEO throughout the Web site.

Table 2 summarises the general behavioural observations that were exhibited throughout the query sessions in respect of: link assessment strategy; user perceptions of organic and sponsored links; and the distribution of first click behaviour. It shows that participants utilised a depth-first approach for each query, however the depth-first approach appeared less deliberate for the transactional query than the informational query as users exhibited heightened awareness when perusing the SERP and processed it more slowly. Sponsored links elicited a far more cynical and negative reaction from participants overall.

Table 2 Behavioural observations from query sessions and verbalisations

	Informational (Query)	**Transactional** (Query)
Organic (Links)	• Heavily distributed on the top of the organic SERP • Deliberate depth-first approach recurrent • SERP was quickly processed • Participants more relaxed	• Heavily distributed on the 1st and 2nd link • A less deliberate depth-first strategy recurrent • SERP processing was much slower • Greater awareness evident
Sponsored (Links)	**N/A**	• Heavily distributed on the 1st and 2nd link • Participant cynicism was evident • Greater anger/frustration exhibited • Less confidence shown in sponsored links

Again, from an SEM perspective, companies should be aware of the implications emerging from this research. This study found some preference for clicking on organic over sponsored links. Furthermore, from verbalisations, participants expressed less faith, at times outright distrust, in sponsored links. In considering a paid search campaign, this study has shown that first click behaviour is carried out wholly on the top sponsored links. Choosing right-hand side sponsored links would appear a near-redundant investment. The anomaly between verbalisations about sponsored links and actual usage patterns is of concern. Some participants appeared confused or uncertain regarding the actual authenticity and purpose of the top sponsored links. While the findings put forward a possible benign explanation about how people theorise they will behave in an espoused theory may differ from how they behave in practice, their theory-in-use, the issue is primarily one for Google to resolve by making more explicit the distinction for users.

References

Argyris C (1980) Inner contradictions of rigorous research. Academic Press, New York

Bar-Ilan J (2007) Manipulating search engine algorithms: the case of Google. J Inf Commun Ethics Soc 5(2/3):155–166

Bar-Ilan J, Levene M, Mat-Hassan M (2006) Methods for evaluating dynamic changes in search engine rankings: a case study. J Doc 62(6):708–729

Barry C, Charleton D (2008) Researching search—a study into search engine marketing practices in Ireland. In: Proceedings of the international conference on e-business, Porto, Portugal. 26–29 July 2008 (CD-ROM)

Browne GJ, Pitts MG, Wetherbe JC (2007) Cognitive stopping rules for terminating information search in online tasks. MIS Q 31(1):89–104

Cho J, Roy S (2004) Impact of search engines on page popularity. In: Proceedings of the 13th international world wide web conference, 17–20 May 2004. ACM Press, New York, pp 20–29

Ciaramita M, Murdock V, Plachouras V (2008) Online learning from click data for sponsored search. In: Proceedings of world wide web conference, Beijing, China, 17–22 May 2008. ACM Press, New York, pp 227–236

Crutcher R (1994) Telling what we know: the use of verbal report methodologies in psychological research. Psychol Sci 5(5):241–244

Hochstotter N, Lewandowski D (2009) What users see—structures in search engine results pages. Inf Sci 179(12):1796–1812

Hotchkiss G, Garrison M, Jensen S (2004) Search engine usage in North America. http://www.enquiro.com/research.asp. Accessed 1 Dec 2008

iProspect (2008) Search engine marketing research: iProspect blended search results study. http://www.iprospect.com/about/researchstudy_2008_blendedsearchresults.htm. Accessed 1 June 2009

Jansen BJ, Resnick M (2005) Examining searcher perceptions of and interactions with sponsored results. In: The workshop on sponsored search auctions at ACM conference on electronic commerce (EC'05), 5–8 June, Vancouver, BC, Canada, pp 1–8

Jansen BJ, Molina PR (2006) The effectiveness of web search engines for retrieving relevant ecommerce links. Inf Process Manage 42:1075–1098

Jansen BJ, Brown A, Resnick M (2007) Factors relating to the decision to click-on a sponsored link. Decision Suport Syst 44(1):46–59

Jansen BJ, Booth DL, Spink A (2008) Determining the informational, navigational, and transactional intent of web queries. Inf Process Manage 44:1251–1266

Joachims T, Granka L, Pan B, Hembrooke H, Gay G (2005) Accurately interpreting clickthrough data as implicit feedback. In: Proceedings of the 28th international conference on research and development in information retrieval, Salvador, Brazil, 15–19 August, SIGIR'05, ACM Press, New York, pp 154–161

Jones R (2008) SEO site structure 101. http://searchenginewatch.com/3632183. Accessed 22 Dec 2008

Keane MT, O'Brien M, Smyth B (2008) Are people biased in their use of search engines? Commun ACM 51:49–52

Klöckner K, Wirschum N, Jameson A (2004) Depth and breadth-first processing of search results list. In: CHI'04 extended abstracts on human factors in computing systems, Vienna, Austria, 24–29 April. ACM Press, New York, pp 1539–1539

Laffey D (2007) Paid search: the innovation that changed the web. Bus Horizons, 50:211–218

Lewandowski D (2008) The retrieval effectiveness of web search engines. Considering results descriptions. J Doc 64(6):915–937

Marable L (2003) False oracles: consumer reaction to learning the truth about how search engines work, results of an ethnographic study, Research Report. Consumer WebWatch, Yonkers

O'Brien M, Keane M (2006) Modelling result—list searching in the world wide web: the role of relevance topologies and trust bias. In: Proceedings of the 28th annual conference of the cognitive science society, Vancouver, Canada, 26–29 July, pp 1881–1886

O'Brien M, Keane MT, Smyth B (2006) Predictive modelling of first-click behaviour in web-search. In: Proceedings of the 15th international conference on world wide web, Edinburgh, Scotland, 23–26 May, ACM Press, New York, pp 1031–1032

Richardson M, Dominowska E, Ragno R (2007) Predicting clicks: estimating the click-through rate for new ads. In: Proceedings of the 16th international conference on world wide web, Banff, AB, Canada, 8–12 May, ACM Press, New York, pp 521–530

Rose, DE, Levinson, D (2004) Understanding user goals in web search. In: Proceedings of the 13th international conference on the world wide web, New York, NY, USA, 17–20 May, ACM, pp 13–19

Szetela D (2008) PPC landing pages: PPC visitors have ADD. http://searchenginewatch.com/3632150. Accessed 19 Dec 2008

Todd M (2006) Getting high traffic from search engines is wasted on poor sites. New Media Age, http://www.mad.co.uk/Main/News/Disciplines/Digital/Industry-Issues/Articles/69252fa4e1f84d5d8f9ea61666e8045e/Getting-high-traffic-from-search-engines-is-wasted-on-poor-sites. html. Accessed 29 June 2008

Van Waes L (1998) Evaluating on-line and off-line searching behavior using thinking-aloud protocols to detect navigation barriers. In: Proceedings of the 16th annual international conference on computer documentation, Quebec, September 1998, pp 180–183

Zhang Y, Jansen BJ, Spink A (2009) Time series analysis of a web search engine transaction log. Inf Process Manage 45(2):230–245

Effective QoS Aware Web Service Composition in Dynamic Environment

Peter Bartalos and Mária Bieliková

Abstract Nowadays, several web services composition approaches, each considering various aspects of the composition problem, are known. Consequently, a lot of attention shifts to the performance issues of web service composition. This paper presents an effective QoS aware composition approach. Our work focuses to its performance, which is studied also in dynamically changing environment, where new services are deployed, some services are removed, or the QoS characteristics change in time. We analyze the impact of the need to manage these changes to the sojourn of the composition query in the system. The experiments show that the proposed composition system is handling these changes effectively and the sojourn time is not significantly affected.

1 Introduction

Web service composition [8] is a process of arranging several web services into workflows. The aim is to bring a utility, which cannot be provided by a single service. Its automation showed to be a challenging task. The research of service composition in last years tends to focus on issues related to QoS [3, 5, 6, 9, 13, 15], pre-/post-conditions [4, 6, 12], user constraints and preferences (e.g. soft constraints) [2, 10, 14], consideration of complex dependencies between services [7, 16].

Several approaches deal with performance and scalability issues of the composition process [3, 5, 6, 9, 12]. These use effective data structures created during preprocessing to speed up the composition. As it is true for the web in general, also web services change in time. New services are deployed, some of them are removed.

P. Bartalos (✉) · M. Bieliková
Faculty of Informatics and Information Technologies
Institute of Informatics and Software Engineering
Slovak University of Technology, Ilkovičova 3, 842 16 Bratislava, Slovakia
e-mail: bartalos@fiit.stuba.sk

M. Bieliková
e-mail: bielik@fiit.stuba.sk

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_8,

Moreover, the non-functional properties of services may change frequently. The aim is to find a composite service with the best global QoS characteristic, computed from the QoS attributes of single services based on aggregation rules [3, 13, 15]. The composition system must flexibly react to these changes. The solution should reflect the current situation in the service environment. If the changes are not managed, it may happen that the designed composition does not use the right services, it includes services which are already not executable, or it is not optimal from QoS point of view. Hence, we do not achieve the satisfaction of the user goal based on which the composition is realized. On the other side, the dynamic changes of services require updating the data structures of the composition system, before we start new composition. Thus, the dynamic changes affect the composition time. To avoid too much delay, the updates must be realized quickly. In this paper we deal with the performance and scalability of service composition operating in dynamic environment. We present our designed algorithms, which were already used in our previous work and extended to more effectively manage the dynamic changes.

2 Related Work

Several approaches automating the web service composition had already been proposed. They showed to be useful when looking for a composition aware of pre-/post-conditions, QoS and satisfying the user constraints, preferences. Promising results were achieved also regarding the performance. There are approaches able to compose services in acceptable time, even if the number of services in the registry rises to some ten thousands. However, only little attention had been given to the research of approaches handling the dynamic changes in the service environment.

As we studied approaches in [5, 9, 12], we found out that there are some issues being common to them all. First, they use pre-processing during which effective data structures are created. These represent the services and the interconnections between them. Another similarity is in the base of the composition process. It is a forward search, starting with the concepts describing the inputs provided in the query (at semantic level). The set of these provided concepts is then incrementally extended. In each iteration, concepts associated with outputs of services having provided inputs are appended. An input is provided if it is associated with a concept, already included in the set of provided concepts. This continues in iterations until all concepts required in the user query are not provided and new concept can be appended. During the iterations, also an acyclic graph of services, depicting all possible solutions of the composition problem, is built. After all required concepts are produced, to remove redundant services, a backward search is performed, starting with services directly producing the outputs required in the user query.

In [12] the authors present a Prolog based, pre-/post-condition aware composition approach. It does not consider the QoS of services. The aim of pre-processing in this approach is to convert the service descriptions into Prolog terms. The semantic relations (e.g. subsumption) are also pre-processed. The built-in indexing scheme facilitates fast composition. The authors deal also with *incremental updates* required when adding a new service. They results show that managing an update requires much more time, than the composition (e.g. 1 vs. 18 ms).

An effective, QoS aware composition approach is presented in [9]. It is based on modified dynamic programming, and a data structure having three parts. The first represents services, the second concepts, the third the mapping of the concepts to service parameters. The composition is effective due a filtering utilized to reduce the search space. The pruning removes services (i) which have no inputs (thus cannot be executed), and (ii) are not optimal from QoS point of view. During forward chaining, when adding new concepts into the set of provided concepts, the aggregated QoS values of services are updated to find the optimal value. Also the provided concepts are related to an optimal QoS value, calculated based on the optimal aggregated QoS value of services which are producing it.

In [5] we had presented a composition approach dealing with performance and scalability issues. The difference between our approach and [9] seems to be minor from the algorithm point of view. Due poor description of the algorithm in [9], we are not able to exactly state the difference. However, it is clear that in [9] different data structures are used to express the services, and their relation to other services and concepts. Moreover, our work deals also with pre-/post-conditions, just as [12]. These aspects of our composition system are described in [4, 6].

3 Service Composition Process

The aim of our composition approach presented in [4] was to *select usable* services. The service is usable if all its inputs are provided thus can be executed. The inputs are provided in the user query, or as an output of another usable service. To select usable services, forward chaining is realized. During it, we also select the best provider for each input of all services, considering QoS. This process becomes a high computation demanding task as the number of services, their inputs, and number of input providers rises. The complexity of the computation is also strongly dependent on the interconnections between the services.

To speed up the *select usable* services process, we propose search space restriction. Its aim is the selection of *unusable services*. A service is unusable, if at least one its input is not provided in the user query, neither as output of ancestor services. The process lies on identification of such services, for which there is at least one input not provided by any available service, i.e. the only case when it is usable is if the respective input is provided in the user query. These services are identified during preprocessing. We call them *user data dependent services.*

The selection of (un)usable service is done by marking a service. The marks are: *UNDEC* if it is undecided if the service is unusable, *USAB* if the service is usable, *UNUSAB* if the service is unusable, *UDD* if the service is user data dependent, *UDDPROV* if the service is *UDD* and all dependent inputs are provided in the query.

A lot of our work had been redesigning the effective data structures used in [4, 5], to be quickly modifiable when the service environment changes. In the following, we present an overview of the data structures to be able to understand the algorithms introduced later. The most important are:

- *AllServices:* a collection of all available services.
- *ConceptProviders:* a collection, where an element is a key-value pair of (i) *concept* and (ii) list of services having an output associated with *concept.*
- *ConceptConsumers:* a collection, where an element is a key-value pair of (i) *concept* and (ii) list of services having an input associated with *concept.*
- *InputDependents:* the same as *ConceptConsumers,* but contains only services marked with *UDD.*
- *UserDataDependents:* a collection of services marked with *UDD.*
- *SuperConcepts:* a collection, where an element is a key-value pair of (i) *concept* and (ii) list of concepts (transitively) subsumed by *concept.*

A service is represented as a complex data structure having several attributes. Its aim is to store information about service's properties and interconnections to ancestor, and successor services. The main attributes are:

- *usability:* stores the mark of the service as presented before.
- *unprovidedInputs:* the number of unprovided user data dependent inputs.
- *inputs:* list of concepts associated with the inputs.
- *inputProviders:* for each input, it stores a list of services providing it.
- *bestQoSProviders:* for each input, it stores a reference to a service which provides it and has the best QoS.
- *successors:* list of successor services.

3.1 Restricting the Search Space

The *select unusable* services process performs as presented in Alg. 1. It can be started only after we select services having all inputs provided in the user query and mark them as *USAB.* This is the part of Alg. 2 at lines 1–12. Then, the process starts evaluating which services can be marked with *UDDPROV* (lines 1–6). For each *provided input* in the user query, we get the *list* of services from *InputDependents,* requiring the respective *provided input.* For each service in the *list,* we decrement the number of unprovided user data dependent inputs. If this number is zero, the service is potentially usable. We cannot be sure because it may have other inputs which do not rely on provision in the user query. This is evaluated later.

Next, we decide which user data dependent services are not usable (lines 7–9). We traverse *UserDataDependents* and if the respective service is not usable, either has not provided all user data dependent inputs, it is marked with *UNUSAB.* Then, we create a list of services still having undecided usability (lines 10–12).

Up to now, we know which user data dependent services are unusable. The (un)usability of these services influences also their successors. If there is a service

Algorithm 1 Find unusable services: Input: *provided inputs*

```
1:  for all provided input do
2:      list = InputDependents.get(provided input);
3:      for all service in list do
4:          decrement service.unprovidedInputs;
5:          if service.unprovidedInputs == 0 then
6:              service.usability = UDDPROV;
7:  for all service in UserDataDependents do
8:      if service.usability ≠ USAB AND service.usability ≠ UDDPROV then
9:          service.usability = UNUSAB;
10: for all service in AllServices do
11:     if service.usability == UNDEC then
12:         toProcess.add(service);
13: while is service to process do
14:     for all service in toProcess which is undecided do
15:         for all input in service.inputs do
16:             isUnprovidedInput = true;
17:             if input provided in user query then
18:                 isUnprovidedInput = false;
19:             else
20:                 providers = service.inputProviders of input;
21:                 for all provider in providers do
22:                     if provider.usability ≠ UNUSAB then
23:                         isUnprovidedInput = false;
24:             if isUnprovidedInput == true then
25:                 service.usability = UNUSAB;
```

having some input for which all providers were marked with *UNUSAB,* it is unusable too. In the rest, we evaluate this (lines 13–25).

While there is a potential service which may be marked unusable, we traverse services from *toProcess.* For each input of the respective service, we check if it is for sure unprovided, which a default assumption is. If it is provided in the user query, then *isUnprovidedInput* is false. If it is not, we traverse all the input provider services. If there is at least one which is not unusable, then *isUnprovidedInput* is false. This case does not mean that the input is provided, we just cannot be sure that it is not. If we are sure that the service has at least one unprovided input, i.e. *isUnprovidedInput* was not changed to false, it is unusable. If we mark some service as unusable, the evaluation starts again. The reason is that the found unusable service affects its successors, which may thus become unusable too.

After the *select unusable* services process finishes, several services were marked as unusable. These cannot be used in the composition. Thus, we restricted the search space and we will not waste time during the composition with several unusable services. Note that we did not find all unusable services. We found each unusable service which is user data dependent, or it is a (indirect) successor of it.

3.2 Discovering Usable Services

The *select usable* services process has two phases, see Alg. 2. First, we select services having all inputs provided in the user query. Second, we realize forward chaining to find other usable services. The first phase starts by collecting services which have some inputs provided in the user query (lines 2, 3). Then, for each potential service we check if it has provided all inputs (lines 5–12). If so, the service is marked as *USAB* and put in *toProcess.*

The second phase (lines 13–30) performs in a loop for all services in *toProcess.* For a respective *service,* we check if there is some provider of its inputs (lines 15–24). During this, we select the provider having the best QoS at the moment. If the service has provided all inputs, it is marked with *USAB* and we recalculate its aggregated QoS based on the best QoS providers. Then, for all its successors, we check if their usability is undecided or if the *service* can improve the aggregated QoS of the respective *successor.* If so, the *successor* is processed too.

Algorithm 2 Discover usable services: Input: *provided inputs*

```
 1: for all provided input do
 2:    services = ConceptConsumers.get(provided input);
 3:    potentialServices.putAll(services);
 4:    inConceptsMap.put(provided input);
 5: for all service in potentialServices do
 6:    allInputProvided = true;
 7:    for all input of service do
 8:       if inConceptsMap.contains(input) == false then
 9:          allInputProvided = false;
10:    if allInputProvided == true then
11:       service.usability = USAB;
12:       toProcess.add(service);
13: while toProcess.size > 0 do
14:    service = toProcess.first;
15:    allInputProvided = true;
16:    for all input in service.inputs do
17:       inProvided = false;
18:       providers = service.inputProviders of input;
19:       for all provider in providers do
20:          if provider.usability == USAB then
```

```
            inProvided = true;
            update best QoS provider;
        if inProvided == false then
            allInputProvided = false;
    if allInputProvided == true then
        service.usability = USAB;
        update aggregated QoS;
        for all successor in service.successors do
            if successor.usability == UNDEC OR service improves aggregated
              QoS of successor then
                toProcess.add(successor);
```

4 Dynamic Aspects of Service Composition

Our composition system can be seen as a queuing system with different type of requests [1]. It works as depicted in Fig. 1. When the system starts, it initializes its data structures based on currently available services. After, it is waiting for update requests, or user queries (1). These are collected in queues and processed regarding the *first infirst out* rule. The updates are managed with higher, *non-preemptive* priority, i.e. the composition is not interrupted if an update request arrives. If an update request arrives (2), i.e. new service is available, some service become unavailable, or the QoS characteristics of some service had changed, we update the affected data structures. When all the changes are managed (3), the system is ready to process new user query. If a new query has already been received, the system processes it immediately (4). Otherwise, it goes to waiting state (5). Here, it again waits for an update request, or composition query (6). The composition is followed by the reinitialization (7). After, we manage update requests, if some had arrived (8). If not, we process a new user query for composition (9), or go to waiting state (10).

The change of service QoS characteristics are managed in constant time, independently on the size of the service repository. It requires only changing the values of the QoS attributes in the object representing the corresponding service. The updates required because some service is made (un)available are divided into two cases. Adding a new service requires creating a new object representing it and discovering its interconnections with other objects. When the service is removed permanently,

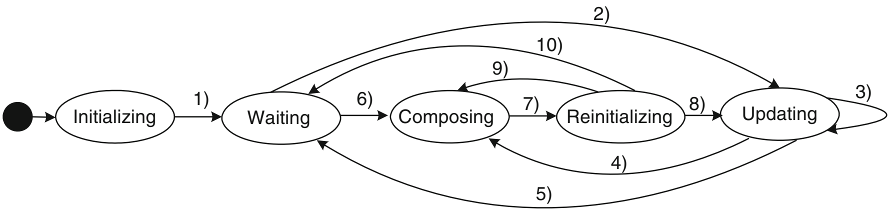


Fig. 1 Composition system life cycle

we have to remove the interconnections with other objects. The temporal removing of the service is possible, too. This is useful if we expect that it will be made available again. The temporal removing of the service is done in constant time, by setting a flag depicting the (un)availability of the service. To make such a service available again, we only have to set the flag once again. The adding of a new service, and permanent removal of some service are more complicated. They depend on the overall set of services available in the service repository, and their interconnections.

The adding of a new service starts with adding a service into *AllServices,* see Alg. 3. Then, we discover the interconnections with ancestor services (lines 2–10). For each input, we determine the services providing it. If there is no input provider, we add the service to *UserDataDependents*, and *InputDependents.* Otherwise, we add it to the list of successors of each provider. Finally, we add it to services requiring the given input.

Algorithm 3 Add service: Input: **service**

```
 1: AllServices.add(service);
 2: for all input in service.inputs do
 3:    providers = ConceptProviders.get(input);
 4:    if providers is empty then
 5:       UserDataDependents.add(service);
 6:       InputDependents.get(input).add(service);
 7:    else
 8:       for all provider in providers do
 9:          provider.successors.add(service);
10:    InputConsumers.get(input).add(service);
11: for all output in service.outputs do
12:    outputSuperConcepts = SuperConcepts.get(output);
13:    for all superConcept in outputSuperConcepts do
14:       allOutputSuperconcepts.put(superConcept);
15: for all superConcept in allOutputSuperConcepts do
16:    ConceptProviders.get(superConcept).add(service);
17:    consumers = ConceptConsumers.get(superConcept);
18:    for all consumer in consumers do
19:       service.successors.add(consumer);
20: for all successor in service.successors do
21:    for all input in successor.inputs do
22:       if allOutputSuperConcepts.contains(input) then
23:          successor.inputProviders.add(service);
24:       InputDependents.get(input).remove(successor);
25:    if UserDataDependents.contains(successor) then
26:       if successor has not unprovided inputs then
27:          UserDataDependents.remove(successor);
```

In the next, we create a collection of distinct concepts subsumed by some service outputs (lines 11–14). For each such *superConcept,* we add the service to a list of its providers (line 16). Each service for which *service* produces input data is added to its successors (lines 17–19). Finally, for each successor we add *service* to providers of all inputs produced by it (line 23). During this, we check if the successor is in *UserDataDependents.* If it is, and the *service* provides an input, which was the only unprovided input, the successor is removed from *UserDataDependents.*

The permanent removal of the services is a revert operation to adding a new service. It deletes the interconnections to other services, and the references of the object representing it, from each data structure. Due a straightforwardness of the process and a lack of space, we omit its detailed description.

5 Evaluation

The evaluation of our approach is split into evaluation of (i) times required to add/remove services, compose services, and reinitialize the system, (ii) behavior of the system due continuing query arrival, without or with dynamic changes in the service registry. All experiments had been realized using a Java implementation of our composition system. The computations had been running on a machine with two 64-bit Quad-Core AMD Opteron(tm) 8354 2.2 Ghz processors with 64 GB RAM.

The experiments were realized using data sets generated by a third party tool (http://ws-challenge.georgetown.edu/wsc09/software.html) used to create data sets for *Web services Challenge 2009* [11]. We generated test sets consisting from 10,000 to 100,000 services. For each test set, the solution requires at least 50 services to compose. The number of concepts in the ontology is from 30,000 to 190,000. To allow comparison with others, the data sets are available at http://semco.fiit.stuba.sk/compositiontestsets.

In Table 1 and Fig. 2 we see how the complexity of the service set (e.g. its size) affects the time required to (i) add a new service (as depicted in Alg. 3), (ii) permanently remove a service, (iii) compose services, and (iv) reinitialize the system after composition. In our experiment we measured the time of adding 1,000

Table 1 Operation times

Web services	Add	Remove	Reinitialization	Composition
10,000	0.84	1.68	1.86	4.95
20,000	0.92	2.84	4.46	14.8
30,000	1.02	4.82	10.2	46.3
40,000	1.53	7.88	13.8	35.9
50,000	1.13	5.81	19.6	37.3
60,000	2.12	10.3	25.6	93.6
70,000	1.39	9.11	27.1	88.0
80,000	1.48	13.3	29.5	64.3
90,000	1.86	9.46	30.0	271.8
100,000	1.89	12.0	51.1	152.4

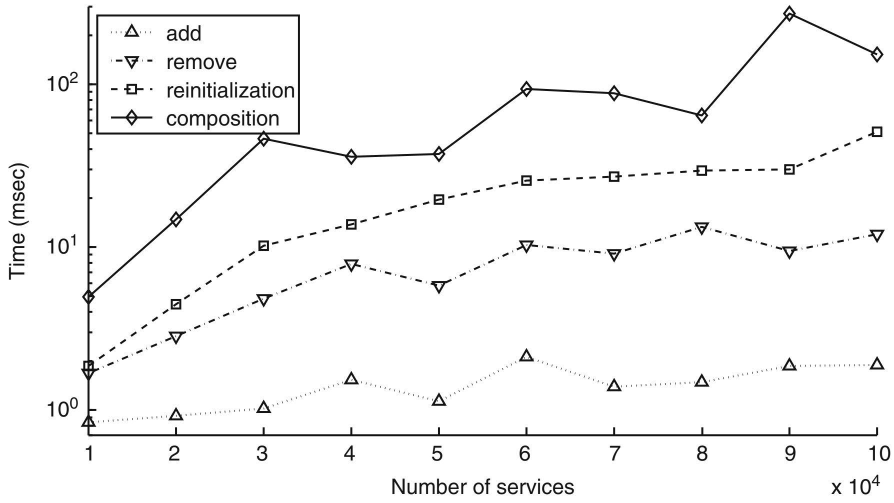


Fig. 2 Add/remove, reinitialization, and composition time

services into a repository, and permanent removing of the same services. As we see, removing a service is more time demanding. This is because the removal requires more traversal of data structures, to find the position of the service's object reference. The composition and update times do not necessary rise as the number of services rises. This is because they are strongly affected also by the interconnections of services, and QoS parameters. The reinitialization time is linearly dependent on the number of all services and the number of user data dependent services. In our experiments, it reached maximum of 53% of the composition time, 33% in average.

In the next we present experimental results considering a composition system as presented in Sect. 4. We measured the sojourn time, and the queue size. The sojourn time is a time between the generation of the update request, or user query and the end of its processing. We assume that the arrivals of both the update requests and user queries are continuous, independent, and occur according to a *Poisson process*. Hence, the interarrival times follow exponential distribution. Since there is no real application, we cannot evaluate these assumptions. Due this, we present also results where the interarrival times follow uniform distribution, to see the effect of different distribution on the measured parameters.

Figures 3 and 4 present the dependency between the mean interarrival time and sojourn time, and mean queue size, for data sets with 20,000 up to 100,000 services. As we see, there is a significant difference between the results when different distributions are used. In the case of exponential, the standard deviation of the results is higher. Moreover, it presents rising tendency as the complexity of the data set, and the sojourn time rises. Similar observations are present also regarding the mean queue size. The system tends to be in stable state when the mean interarrival time is more than a double of the composition time. In this case, the mean queue size

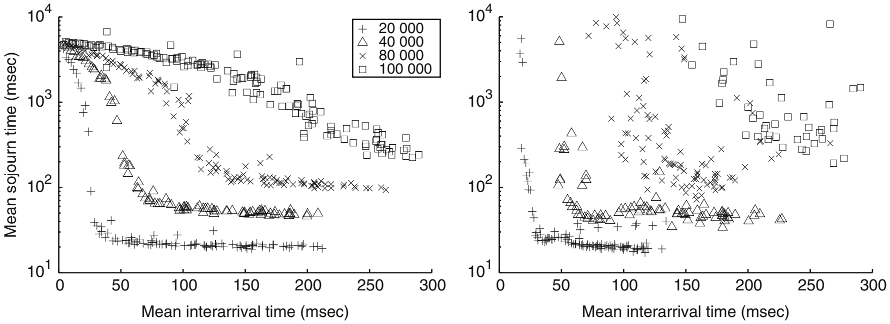


Fig. 3 Sojourn time: uniform distribution at *left*, exponential distribution at *right*

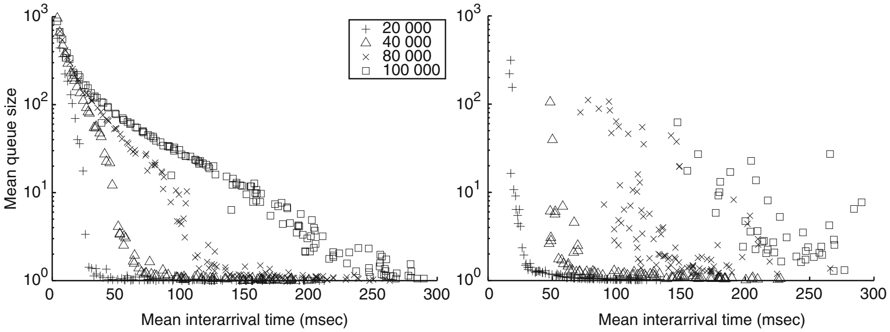


Fig. 4 Queue size: uniform distribution at *left*, exponential distribution at *right*

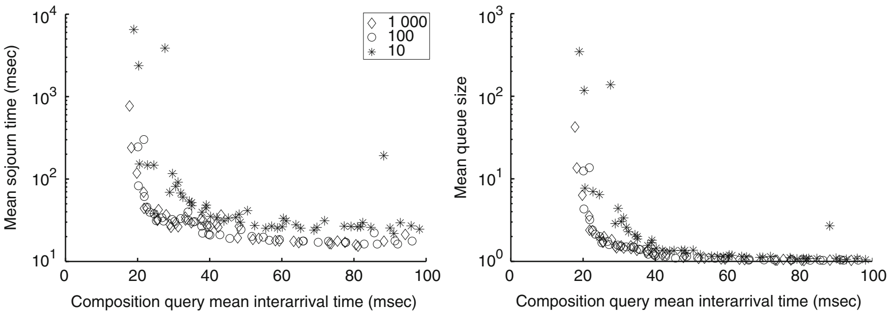


Fig. 5 Effect of dynamic changes in the service environment

(measured after the request is put in it, i.e. it cannot be less than 1) is less than 2. Adequately, the sojourn time is less than a double of the composition time.

Figure 5 presents the effect of the dynamic changes in the service environment. We had been simulating arrival of requests to add a new, or permanently remove a service, for various interarrival times with exponential distribution. We used the test set with 20,000 services. The experiments show that the sojourn time is not significantly enlarged, even if the update requests are frequent. We had dramatically decreased the interarrival time of update requests, from 1,000 to 10 ms. Even in this case, we observed only a little delay in composition. The mean composition query queue size remained low and the stability of the system was not upset.

6 Conclusions

This paper presents a QoS aware composition approach focusing on efficiency, and scalability in dynamically changing service environment. We adapted our previous approach to effectively handle the update requirements. The previous approach proved to be fast as it was successful at *Web services Challenge 2009,* a world competition aimed at well performing service composition [11].

Our composition system manages the updates in much shorter time than composition queries (about one order of magnitude). We have investigated its behavior when simulating continual arrival of user queries collected in queues. Our experiments show that if the mean interarrival time of composition queries is more than a double of composition time, the sojourn time is not significantly higher. When the queries arrive more frequently, the sojourn time rises dramatically. Since our composition system manages effectively the update requests, these do not significantly affect the sojourn time. We have made also a simulation model of our composition system. It is an effective tool to analyze its behavior, without the need of running it.

Our future work is to study the behavior of the composition system regarding new simulation scenarios, such as arrival of batches of update requirements. We will study what are the bottlenecks of the composition system, and what modifications could overcome them. Our assumption is that for example parallel processing of user queries significantly reduces the sojourn time. We will use also a simulation model of our composition system, to analyze what modifications are useful.

Acknowledgements This work was supported by the Scientific Grant Agency of SR, grant No. VG1/0508/09 and it is a partial result of the Research & Development Operational Program for the project Support of Center of Excellence for Smart Technologies, Systems and Services II, ITMS 25240120029, co-funded by ERDF.

References

1. Adan I, Resing J (2002) Queueing theory. Eindhoven University of Technology, 180 pages http://www.win.tue.nl/%7Eiadan/queueing.pdf. Accessed Apr 2010
2. Agarwal S., Lamparter S (2005) User preference based automated selection of web service compositions. In: Proceedings of the international conference on service-oriented computing 2005, workshop on dynamic web processes, Amsterdam, pp 1–12
3. Alrifai M, Risse T, Dolog P, Nejdl W (2009) A scalable approach for qos-based web service selection. In: Proceedings of the international conference on service-oriented computing 2008 workshops, Springer, Berlin, pp 190–199

4. Bartalos P, Bielikova M (2009) Fast and scalable semantic web service composition approach considering complex pre/postconditions. In: Proceedings of the 2009 IEEE congress on services, international workshop on web service composition and adaptation, IEEE CS, Washington, DC, pp 414–421
5. Bartalos P, Bielikova M (2009) Semantic web service composition framework based on parallel processing. In: Proceedings of the international conference on e-commerce technology, IEEE CS, Washington, DC, pp 495–498
6. Bartalos P, Bielikova M (2010) QoS aware semantic web service composition approach considering pre/postconditions. In: Proceedings of the international conference on web services. IEEE CS, Washington, DC, pp. 345–352. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5552768&tag=1
7. Bertoli P, Kazhamiakin R, Paolucci M, Pistore M, Raik H, Wagner M (2009) Control flow requirements for automated service composition. In: Proceedings of the international conference on web services, IEEE CS, Washington, DC, pp 17–24
8. Dustdar S, Schreiner W (2005) A survey on web services composition. Int J Web Grid Serv 1(1):1–30
9. Huang Z, Jiang W, Hu S, Liu Z (2009) Effective pruning algorithm for qos-aware service composition. In: Proceedings of the international conference on e-commerce technology, IEEE CS, Washington, DC, pp 519–522
10. Karakoc E, Senkul P (2009) Composing semantic web services under constraints. Expert Syst Appl 36(8):11021–11029
11. Kona S, Bansal A, Blake MB, Bleul S, Weise T (2009) Wsc-2009: a quality of service-oriented web services challenge. In: Proceedings of the international conference on e-commerce technology, IEEE CS, Washington, DC, pp 487–490
12. Kona S, Bansal A, Gupta G (2007) Automatic composition of semantic web services. In: Proceedings of the international conference on web services, IEEE CS, Washington, DC, pp 150–158
13. Lecue F., Mehandjiev N (2009) Towards scalability of quality driven semantic web service composition. In: Proceedings of the international conference on web services, IEEE CS, Washington, DC, pp 469–476
14. Lin N, Kuter U, Sirin E (2008) Web service composition with user preferences. In: Proceedings of the European semantic web conference, Springer, Berlin, pp 629–643
15. Rosenberg F, Celikovic P, Michlmayr A, Leitner P, Dustdar S (2009) An end-to-end approach for QoS-aware service composition. In: Proceedings of the international enterprise distributed object computing conference, IEEE, Washington, DC, pp 151–160
16. Yu HQ, Reiff-Marganiec S (2009) A backwards composition context based service selection approach for service composition. In: Proceedings of the international conference on services computing, IEEE CS, Washington, DC, pp 419–426

The Right Tools for the Right Tasks: Meaningful Education for IS Professional

Peter M. Bednar and Christine Welch

Abstract The predicament in which we find ourselves today is that many professionals lack a sufficient grounding in formal methods, tools and techniques to enable them to make an appropriate selection for different kinds of problem. This can mean that complexity remains unrecognized and ambiguous problem situations are addressed as if they were clear and straightforward, resulting in inadequate solutions that are not experienced as useful by clients. We are thus faced with a circular dilemma. Those who attempt to use, e.g. SSM are unable to do so effectively through lack of understanding. They are thus driven back to the need for formal methods, and the disadvantages inherent in these approaches which SSM was originally created to address. Thus, there is a need to reintroduce into the agenda of soft and Agile methods an understanding of the skills and tool sets offered by hard/formal approaches. New professionals require a comprehensive *education* in use of tools and techniques, including their complementarity. This will not be delivered by training individuals in application of particular methodologies in a piecemeal and fragmented way, but by thorough and rigorous examination of whole methodologies in use. Only then can they engage in practice in the real world and develop their own tool sets, from which to select in an informed way those most appropriate to a problem situation.

1 Introduction

For many years, hard systems methodologies have been criticized for being unwieldy, prescriptive and time-consuming. Furthermore, these approaches ignore the need to explore the problem space and develop it taking into account the perspectives of different engaged actors (see Checkland 1999; Mumford 1995). Reactions against this have led to such diverse developments such as the Soft Systems Methodology

P. M. Bednar (✉)
School of Computing, University of Portsmouth, Portsmouth, UK
e-mail: peter.bednar@ics.lu.se

Department of Informatics, Lund University, Lund, Sweden

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_9,

Checkland (1999), which attempted to incorporate the bigger picture, or prototyping (Smith 1991) and Agile methods (Shore and Warden 2008; Mumford et al. 2006), which attempted to support rapid progress towards solutions. Agile methods can be considered a response to the perceived disadvantages of formal methodologies, which could produce over-engineered solutions (Avison and Fitzgerald 2006). Recognizing that the World changes very rapidly, Agile methods are intended to achieve a solution that may be imperfect but is achieved while still relevant to current problems. Agile methods move quickly from one aspect to another without necessarily attempting permanent and lasting solutions.

Initial critique of formal methodologies (e.g. SSADM and JSD and their predecessors) came from academics and professionals well-versed in the tools, techniques and application of such approaches (see e.g. Langefors 1966; Avison and Wood-Harper 1990). Flensburg (2008) comments that early developments in organizational Informatics followed in the wake of approaches to Computer Science that were themselves driven by a focus on numerical analysis (see Flensburg 2008).

These critics were well qualified to create ideas for new approaches that might overcome the disadvantages of structured methods. However, less skilled practitioners have subsequently made less-than-informed choices about approaches to development. These were not always successful because of uninformed ideas about inherent flaws in formal methods, combined with an imperfect understanding of the practical strengths and weaknesses of either category. As a result, soft methods were also criticized as unsatisfactory in use, due to similar ignorance of any tools that would render these approaches productive (see, e.g. Kreher's study (1994)). However, what is actually lacking is a sufficiently deep understanding of the nature of systemic analysis and design principles.

Thus, soft and Agile approaches have both, in their turn, been subject to criticism for failing to provide a blueprint from which to build a new system. Nevertheless, analysts only familiar with Agile approaches, without a deep understanding of first principles, may try to apply them regardless of context, leading to poor design, inconsistency and low traceability (Avison and Fitzgerald 2006). The predicament in which we find ourselves today is that many professionals lack a sufficient grounding in formal methods, tools and techniques to enable them to make an appropriate selection for different kinds of problem. This can mean that complexity remains unrecognised and ambiguous problem situations are addressed as if they were clear and straightforward, resulting in inadequate solutions that are not experienced as useful by clients (Avison and Wood-Harper 1990). New professionals require a comprehensive *education* in use of tools and techniques, including their complementarity (see Mathiassen and Purao 2002; Omland 2009). This will not be delivered by training individuals in application of particular methodologies in a piecemeal and fragmented way, but by thorough and rigorous examination of whole methodologies in use. Only then can they engage in practice in the real world and develop their own tool sets, from which to select in an informed way those most appropriate to a problem situation.

2 Educating the IS Professional

Omland (2009) discusses an indissoluble relationship between competence, methods and practice. Reporting research into a successful ISD in a Norwegian municipality, he states:

> In theory, competence, methods and practice are separate and clearly distinct elements. In actual ISD, however, the three elements form close and integrated relationships. (Omland 2009, p. 3)

Mathiassen et al. (2000) discuss teaching as a process in which students must gain basic skills from which to build their understandings. Their pedagogical approach to Object Oriented analysis and design goes through the approach step-by-step. However, it is necessary to understand that application does not follow the linear path of the pedagogical vehicle, but reflects the complexity of context in which it is applied.

It can be seen, therefore, that a professional analyst requires more than just knowledge of the steps involved in any particular methodology. It is necessary to combine such knowledge with practical skill and the ability to reflect upon a problem space in order to select and apply *relevant* tools. Furthermore, a problem is neither given nor predictable; it emerges and changes during exploration. Analysts need to reframe the problem as they proceed. Thus, analysis is *contextual* (Checkland 1999; Bednar 2000). This in itself presents a dilemma: analysts cannot know what to explore in the context until they have explored it. However, that means that they do not know what is to be explored. Furthermore, deep understanding of contextual dependencies is vested in the engaged actors who participate in the context under examination. It is easy to say "consult with stakeholders" but another matter entirely to choose with whom to liaise, under what conditions and when. The internal politics of situations often constrain who can be involved and how they participate. Contextual analysis is messy and problematic. Even a skilled and experienced professional analyst does not automatically know, in a particular context, what questions to ask and of whom.

It has long been recognized that there is a need for a sociotechnical approach to design, i.e. that the development of a technical solution will be inadequate without consideration of the work practices and human context within which such a solution will be implemented (Cherns 1976).

Analysis and design are more commonly undertaken in order to change a legacy system than to embark on development of a completely novel IS. This leads us to reflect that approaches to education of IS professionals that discuss methodologies as if they are to be applied as recipes, creating systems from scratch on a step-by-step basis, will be an inadequate preparation for the exigencies of professional life, engaging in maintenance, enhancement and re-engineering of existing systems. Such a view is confirmed by Omland (2009), and Mathiassen et al. (2000) and by Madsen et al. (2006). Analysts must collaborate with engaged actors to explore problem situations in order to support creation and shaping of requirements. It is essential that a naive and premature view of "the problem" is avoided in favour of an holistic and exploratory approach. Differences in perspective among engaged actors (highlighted by Checkland (1999) in relation to soft systems approaches) must be taken into account in questioning what the nature of 'the problem' may be. Ulrich (2001) points out a need to consider the stance from which problem definition is undertaken, and undertake *boundary critique*—what is considered within the scope of inquiry and what is excluded is an important decision for the success of any design process and can only be undertaken in collaboration with engaged actors. It is these individuals for whom any created system must be perceived as useful, and it is they who must therefore own and control the context of development (Friis 1991).

Since competence, methods, and practice can be seen to form a tight, integrated whole (Omland 2009), it follows that new practitioners cannot gain sufficient understandings from an academic and detached study of particular methodological 'recipes'. It is vital for educators to promote an engaged attitude among students of analysis and design. Brown et al. (1997) report the results of a series of interviews in which students were invited to explain what they understood by "learning". A range of definitions emerged:

- Learning as an increase in knowledge. These students apparently saw themselves as acquiring this "commodity" from their tutors;
- Learning as memorizing. Here students appeared to see their task as storage of the said "commodity" for a temporary period;
- Learning as acquiring facts or procedures which to be used—skills, algorithms, formulae. These they appeared to see as means to particular ends, e.g. as background to later material or for use in an examination;
- Learning as making sense. These students appeared to make active attempts to abstract meaning in a process of learning so that they could describe their methods and reasoning, as well as answers to problems or tasks;
- Learning as understanding "reality". This group of students appeared to see learning as personally meaningful. They described a transformation in perceptions of the world before and after learning.

It is clear from the discussion above that an educational experience matching student views 1–3 above would provide an inadequate preparation for a student embarking upon professional practice. The pedagogical approach adopted must provide maximum opportunity for students to engage their own sense-making processes and to apply what they learn to their own (current or future) professional roles. They require encouragement to engage in reflection, entering into creation of productive learning spirals. At this point, it is useful to consider the possibility of multiple orders of learning (see, for example, Bateson's discussion (1972, p. 287)). Argyris and Schon (1978) describe two distinct orders of learning in terms of single- or double-loop learning. When an individual needs to solve an immediate problem, she may harness her sense-making processes in order to close a perceived gap between expected and actual experience. In doing so, she operates within a context of existing goals, values, plans and rules, without questioning their appropriateness. However, if she goes beyond this to engage in reflection, and in doing so challenge and critically appraise the assumptions previously applied, this may be considered as double-loop learning.

3 Notes from the Field

The descriptions which follow show how this challenge has been taken up in the authors' own experience of educating undergraduates preparing for careers as IS professionals in a UK University. For a number of years, results in a Level 2 Unit in Systems Analysis and Design had been very disappointing. The pedagogical style of this Unit was didactic, introducing modeling techniques and giving an overview of various methodologies. It was assessed primarily by examination. Reflection among the course team identified lack of student engagement with 'real world' contexts as a possible cause of the unacceptable failure rate. Anecdotally, a student had commented to his tutor during a modeling seminar that 'he quite enjoyed doing these puzzles but had no ideas what they were supposed to be for'; he didn't understand the purpose of systems analysis. New Units were developed in response to these reflections, in an attempt to address this perceived problem.

Level 2 Unit (building on introductory work at Level 1) was designed to include a curriculum and practical tasks that engage students with difficulties inherent in analysis and modeling techniques in a real world context. 114 students took this unit in 2007; 62 in 2008 and 92 in 2009. The approach is grounded in an updated version of the ETHICS methodology (Cherns 1976; Mumford 1983, 2003). Each student is asked to analyse a different organization, gathering the data for themselves by visiting their chosen context. In 2007 these were drawn from local supermarkets; in 2008 students analysed local GP practices; and in 2009 local pharmacies. Templates have been produced covering approximately 30 different analyses (e.g. business process analysis, work design, social analysis), each of which incorporates a guide to application. In applying each of these, students are required to proceed by adopting different perspectives, making use of a range of techniques. The results achieved at first attempt were over 75% in 2007; 70% in 2008 and 68% in 2009, comparing very favorably with earlier versions for which pass rates had fallen as low as 25%.

In the Level 3 unit, the Soft Systems Methodology (Checkland 1999) was chosen as a vehicle to encourage appreciation of the messy and problematic nature of organizational contexts for analysis. The object is for students to engage with the benefits of SSM in conjunction with application of a chosen toolsets from other methodologies. This is similar to the approach taken by Checkland in first introducing SSM to professionals experienced in hard/formal methods (Checkland 1999). The Unit encourages students to in engage with creation of a systemic learning process, using problem-based and problem-focused learning experiences, and the pedagogic style is Socratic method (Engel 1991; Jarvis 2006). Students are given an unstructured problem scenario and asked to develop the problem space and create the boundary from different perspectives. Different categories of skills/tools are then applied selectively depending on how their inquiry develops the problem space. Supporting lectures focus on problematic issues relating to application of hard/formal methods. The intention is to promote understanding about assumptions underlying professional practice. Students apply analytical techniques from methodologies such as, e.g. SSM (Checkland 1999), and Client-led Design (Stowell and West 1995) e.g. Rich Pictures, mind mapping, CATWOE (Checkland 1999), or PEArL (Champion and Stowell 2001), FACTOR (Mathiassen et al. 2000). A range of seminar themes are designed to introduce students to complex and ambiguous problem spaces that have no simple, straightforward answer. Students are given a common objective for which they must create (individually) a suitable methodology. They then develop and apply a framework for evaluation that they can justify as relevant in order to analyze the strengths and weaknesses of their chosen approach. Thus, the focus of this unit is on relevance, and in particular boundary setting and critique. Students must recognize that systems are not given—they exist only as mental constructs depending upon perspectives of individual observers engaged in a learning-based inquiry. In 2007, 13 out of 14 submissions achieved pass marks. In 2008, 22 out of 24 submissions were successful at the first attempt and in 2009, 17 out of 20. These results suggest that student engagement was supported by the approach taken.

Thus, at Level 2, students are introduced to a particular tool kit and given opportunities to experience it in use. At Level 3, fuzzier problems are introduced. Students must explore the problem space for themselves, set and question the boundaries of the problem they wish to explore, using Ulrich's concept of boundary critique (Ulrich 2001). They must apply tools that are more ambiguous in use and require creative thinking, judgment and selectivity.

4 Lessons Learned

Reflection suggests that the new pedagogical approach at Level 2 engaged students with a lot of questions about detail and process of inquiry. Challenging lessons for students related to those of real world professional practice, including: use of language and imitation; a need for reflection upon experience—we cannot think (cognise) about phenomena to be analysed unless we can first recognise them; a need to exercise imagination and recognition in order to surface preconceptions and prejudices. Students need to understand that the complexity of rich, organizational contexts requires complexity in approaches to analysis. Attempts to simplify context are unlikely to inform useful practice. They need to explore links between thinking processes and ways in which thoughts are then represented in reporting and developing resolutions. Most importantly, students need to appreciate that professional analysts cannot analyse contexts and/or develop solutions 'for' stakeholders. Involvement of interested actors, leading to dialogue and description of problem spaces, is essential.

At Level 3, a lot of questions are raised for students about a focus on relevance in inquiry (not just rigour) and creation of a useful process of inquiry to support useful analysis in context. Challenging lessons for students include: subjective selection and choices; subjective moral and ethical judgements; the relationships between analysis and evaluation, context and relevance; developing processes for reflection; subjective validity setting and system boundary questions.

5 Summary and Conclusion

How should an IS professional be prepared to undertake development work? Flensburg (2008) and Mathiassen (2000), among others, criticise pedagogical approaches that treat methodologies as recipes—students need to be able to understand how to follow the recipe but real educational value is in learning how not to follow it—methodology is not prescriptive but enabling. First, it is necessary to know of the existence and potential context of application of a particular tool in the toolbox. To be able to understand any tools, it is necessary to understand the context in which such a tool will be useful—to be able to judge its relevance. Flensburg, in particular, criticises teaching future IS professionals as if the only developments they will face are those undertaken in-house where the process starts from scratch and is pursued through application of a whole methodology—SSADM, RAD or whatever—from beginning to end. He points out that real world experience of IS development is far more likely to focus on maintenance and redesign of systems. This becomes clear when we consider that many companies today adopt service-oriented rather than developmental practice. An example can be seen in the case of Skandia, who two years ago outsourced much of their technical development and maintenance work and diverted their resources to employing business analysts, rather than technical experts (Grant 2007). In a recent White Paper, IBM emphasised a need for more professional IT Service Managers with a deep understanding of the business and its processes in order to facilitate meaningful discussions with business colleagues about their needs (Salvage and Dhanda 2007). Thus, analysts employed in future will increasingly be engaged in continuous improvement of business processes, not development of technical solutions from inception to implementation. It is vital therefore that educational programmes are designed to reflect this, discussing not development of IT but re-development of organizational processes incorporating IT (which may or may not include new IT systems or artefacts). For these reasons, the units described here focus upon information and context, rather than the technologies by which data is processed. This is the essence of the distinction between computer science and informatics (Langefors 1966).

A person who is ignorant of the role and context in which application of particular tools is appropriate will be unable to use them except according to particular rules and instructions. Judgment in the use of tools requires a deeper understanding.

This is apparent from, e.g. Ciborra's (2002) discussion of the role of bricolage in development of useful systems—improvisation can only take place from a platform of skill and understanding in the principles, tools and techniques upon which good design is founded, i.e. informed judgment.

Rigor in professional practice is of course vital. To be able to select and apply tools, you first need to learn about the "correct" role and application of those tools. For example, a trainee bricklayer will first learn to build small walls in a workshop environment. The walls are not intended to serve any purpose but she/he learns about the integrity of walls, including the skills in the use of a trowel, correct mixing of mortar, and correct alignment of bricks. Every type of skilful professional activity has its own standards of rigour, relating to the purpose of that activity and to quality choices. Appreciation and assessment of risk is another important factor here. Relevance emerges in selection of an appropriate tool for the kind of work to be undertaken in context. Possibly, adaptation or invention of an appropriate tool will be needed where none exists for a particular purpose. Again, ability to make informed judgments is critical here. Understanding how scale and complexity impact upon context and therefore choice of appropriate tools and techniques is a further aspect of relevance in practice—the greater the consequences of failure, the greater the importance of attention to relevance. A professional needs to know when selection of relatively minor items can be crucial to overall success. Thus, a rope to a rock climber may not be expensive but his life may depend upon selection of the right for his purpose.

What are we educating people for—relevance or rigor? Is it meaningful to educate people in methodological toolkits? On the one hand, methodologies have been developed in order to "professionalise" the approach taken to systems analysis and development—focusing on rigour. On the other hand, methodologies have been criticised as being naïve, restrictive, unwieldy, etc.—not supportive of relevance. Fashion can have an impact in influencing professional education—methodologies come into and go out of vogue with no relation to their actual usefulness to professionals in context. The past twenty years have seen a move from structured to OO methods. Structured methods were developed as a response to the perceived inadequacy of ad hoc approaches. OO was then promoted because of perceived inadequacies in solutions 'not designed'. Thus, professionals (and their educators) may have attempted to solve a problem of lack of skill in application of tools by substituting a different tool—the problem, however, persists. Educators may have focused on drilling students in particular techniques within a methodology, without providing them with expertise in application of the toolkit to a contextual problem space.

References

Argyris C, Schön DA (1978) Organisational learning. Addison Wesley, Reading

Avison DE, Fitzgerald G (2006) Information systems development, 4th edn. McGraw-Hill, Maidenhead

Avison DE, Wood-Harper AT (1990) Multiview: an exploration in information systems development. McGraw-Hill, Maidenhead

Bateson G (1972) Steps to an ecology of mind. Chicago University, Chicago

Bednar PM (2000) A contextual integration of individual and organizational learning perspectives as part of IS analysis. Inf Sci 3(3):2000

Brown G, Bull J, Pendlebury M (1997) Assessing student learning in higher education. Routledge, London

Champion D, Stowell FA (2001) PEArL: a systems approach to demonstrating authenticity in information system design. J Inf Technol 16:3–12

Checkland P (1999) Systems thinking, systems practice: a 30-year retrospective. Wiley, Chichester

Cherns A (1976) Principles of socio-technical design. Hum Relat 2:783–792

Ciborra CU (2002) The labyrinths of information. Oxford University Press, Oxford

Engel CE (1991) Not just a method but a way of learning. In: Boud D, Feletti G (eds) The challenge of problem-based learning. Kogan Page, London, pp 23–33

Flensburg P (2008) The new informatics revolution—not. IRIS 31, Åre, Sweden, 10–13 Aug 2008

Friis S (1991) User controlled information systems development. Departement of Information and Computer Science, Lund University, Lund

Grant I (2007) India deal clears Skandia's app maintenance backlog. Comput Weekly, 6(Nov):4

Jarvis P (2006) The Socratic method. In: Jarvis P (ed) The theory, 2nd edn. Routledge, London, pp 90–97 (July 2006)

Kreher H (1994) Some recurring themes in using SSM. J Oper Res Soc 45:1293–1303

Langefors B (1966) Theoretical analysis of information systems. Studentlitteratur, Lund

Madsen S, Kautz K, Vidgen R (2006) A framework for understanding how a unique and local development method emerges in practice. EJIS 15:225–238

Mathiassen L, Munk-Madsen A, Nielsen PA, Stage J (2000) Object-oriented analysis and design. Forlaget Marko, Aalborg

Mathiassen L, Purao S (2002) Educating reflective systems developers. Inf Syst J 12(2):81–102

Mumford E (1983) Designing Human Systems: For New Technology. Manchester Business School

Mumford E (1995) Effective Systems Design and Requirements Analysis: The Ethics Method. Palgrave

Mumford E (2003) Redesigning human systems. IRM Press, Hershey

Mumford E, Hickey S, Matthies H (2006) Designing human systems: an agile update to ETHICS. Lulu.com, Raleigh

Omland HO (2009) The relationships between competence, methods, and practice in information systems development. Scand J Inf Syst 21(2):3–26

Salvage I, Dhanda IS (2007) IBM white paper. IT service management: is it now too important to leave to the IT department alone? How and why the IT function needs to change its relationship with the business. IBM Global Technology Services, Sept 2007. http://whitepapers.theregister.co.uk/search/?q=IBM. Accessed 11 Nov 2007

Smith MF (1991) Software prototyping: adoption, practice and management. McGraw-Hill, London

Shore J, Warden S (2008) The art of agile development. O'Reilly Media Inc., Sebastopol

Stowell F, West D (1995) Client-led design: a systemic approach to information systems definition. McGraw-Hill, Maidenhead

Ulrich W (2001) Boundary critique. In: Daellenbach HG, Flood, RL (eds) The informed student guide to management science, 2002. Thomson Learning, London

Conditions Influencing Client–IS Supplier Interactions During IS Outsourcing

Linda Bergkvist

Abstract The extent of outsourcing information systems (IS) activities has been significant and the trend seems likely to continue in the foreseeable future. IS outsourcing relationships bring challenges that need to be understood and managed. The client–IS supplier relationship as such is one of the less frequently examined topics in IS outsourcing research. This paper contributes to the IS outsourcing relationship literature by describing conditions influencing client–IS supplier interactions during IS outsourcing. The paper sets out from the interaction approach as it contributes new insights to IS outsourcing relationships. The interaction approach is complemented with elements to increase the description of IS outsourcing relationship interactions. The elements are composed of viewing the client firm and the IS supplier firm as being comprised of business, process and IS level and the dimensions contract and management. The conditions influencing client–IS supplier interactions are presented in the form of a conceptual framework which when adopted can contribute to a better understanding of the features of IS outsourcing relationships and facilitate for firms in their decision-making of IS outsourcing.

1 Introduction and Purpose

Outsourcing of information systems (IS) has been one of the most discussed phenomena within the IS field in recent years. The extent of outsourcing IS activities has been significant and the trend seems likely to continue in the foreseeable future. IS activities include activities related to the IS life cycle, such as IS development and IS maintenance management. The relationship between client and IS supplier is commonly captured by the generic IS outsourcing term, which is referring to both globally and domestically client–IS supplier relationships. The success of client–IS supplier relationships has been argued to effect IS outsourcing outcomes positively (King and Torkzadeh 2008). A recent study shows that several IS outsourcing initiatives fail to fulfil expectations on business performance improvements (Dabhilkar

L. Bergkvist (✉)
Department of Information Systems & Project Management, Karlstad University, Karlstad, Sweden
e-mail: linda.bergkvist@kau.se

J. Pokorny et al. (eds.), *Information Systems Development*, 125
DOI 10.1007/978-1-4419-9790-6_10,

et al. 2009). Among the issues discussed are the failure of managing IS outsourcing relationships and the lack of parties' mutual understanding of each other's businesses. These findings from the literature together with the argument that interactions are focal events in IS outsourcing relationships (Kern and Willcocks 2000) motivate the use of a relationship perspective for studying IS outsourcing. This is further motivated as the actor perspective has dominated prior research on IS outsourcing (e.g. Bhatt et al. 2006). By *relationship perspective* is in this paper meant the client firm and the IS supplier firm and their intra- and inter-firm interactions (cf. Lacity and Willcocks 2000). The definition of an IS outsourcing relationship used in this paper is adopted by Goles et al. (2005, p. 49):

> an ongoing, long-term linkage between an outsourcing vendor and customer arising from a contractual agreement to provide one or more comprehensive IS activities, processes, or services with the understanding that the benefits attained by each firm are at least in part dependent on the other.

A thorough examination of IS outsourcing relationships has been requested for in prior research (Kern and Willcocks 2002). The desire for emphasising IS outsourcing relationships is related to firms' underestimation of the efforts necessary to reach a successful relationship. Firms need to be aware of the close attention that must be paid to aspects related to client–IS supplier relationships. This includes for example the criteria for selecting an IS supplier, the details included in the contract and the monitoring of the outsourcing process. Research studies emphasising the client–IS supplier relationship have above all contributed findings related to the contractual arrangement between client and IS supplier (Gonzales et al. 2006). The client–IS supplier relationship as such and what constitutes the key success conditions in IS outsourcing are less frequently examined topics in IS outsourcing research (Alsudairi and Dwivedi 2010). To address the identified knowledge gap this paper focuses features related to IS outsourcing relationships by addressing the following purpose: *to describe conditions influencing client–IS supplier interactions during IS outsourcing*.

2 Literature Review

Several theories have been used to study IS outsourcing and some of the more popular ones are Transaction Cost (TC) Theory, Agency Theory and Resource-Based Theory. The TC Theory has been well applied in the IS outsourcing research to explain the decision for IS outsourcing, which also is the case of Agency Theory and Resource-Based Theory (King and Torkzadeh 2008). Since outsourcing requires the building of relationships theories focusing interorganisational relationships are motivated when studying IS outsourcing. Theories such as the Relationship Exchange Theory (RET) and Psychological Contract Theory, which are using the relationship as the unit of analysis, provide valuable insights for studying IS outsourcing relationships. Moreover, Industrial marketing and business literature focuses on interorganisational relationships. One field within the Industrial marketing literature is the interaction approach which contributes broad and deep knowledge on business interaction (Axelsson 2010). The interaction approach has proved robust enough to be applied to a wide range of different interorganisational relationships such as interorganisational relationships in IT and IS outsourcing contexts (e.g. Leek et al. 2000; Ekman 2006). The interaction approach moreover uses both the organisational and the individual level, which has been found valuable for describing and explaining the exchanges in IS outsourcing relationships (Kern and Willcocks 2001).

In summary, the review on IS outsourcing literature shows that few studies have applied the client–IS supplier relationship as the studied subject. Furthermore, the review shows that interorganisational relationship theories are not represented among the most frequently used theories in IS outsourcing research (Alsudairi and Dwivedi 2010). The application of the interaction approach therefore contributes new insights and knowledge to the description and explanation of IS outsourcing relationships. The interaction approach emphasises that both client and IS supplier should be seen as active parties and should therefore be given equal attention. Moreover exchanges and interactions are argued to be the focal events in IS outsourcing relationships (Kern and Willcocks 2000), which coincides with the basic ideas of the interaction approach. Finally, practitioners have argued that the interaction approach is a favourable approach for analyzing specific business contexts (Axelsson 2010), such as IS outsourcing contexts.

3 The Interaction Approach

This section begins with an introduction of the interaction approach. Thereafter the identified limitations of the interaction approach for studying IS outsourcing relationships are presented together with how they are addressed in this research study.

3.1 Introducing the Interaction Approach

The interaction approach focuses the interaction process which ties client and supplier together. The main elements of the interaction approach are illustrated in Fig. 1. Four dimensions describe and influence the interactions between client and supplier (Håkansson 1982):

- the elements of the interaction process,
- the actors involved, both as organisations and individuals,
- the environment in which the interaction takes place and
- the atmosphere that influences on, and is influenced by, the interaction.

The interactions are described in terms of *short-term exchange episodes* and *long-term processes*. The short-term exchanges consist of four exchange episodes: (1) product/service exchange, (2) information exchange, (3) financial exchange and

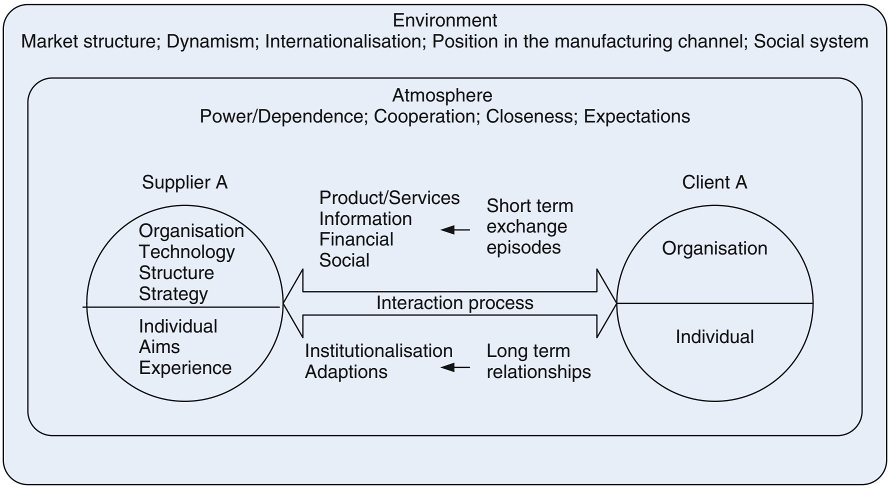


Fig. 1 The interaction model. (Håkansson 1982, p. 24)

(4) social exchange. The long-term exchange process is characterised by institutionalisation and adaptation of processes and procedures as part of long-term relationships. The long-term exchanges are a result of interactions that have occurred between client and supplier over a long time period and have become institutionalised. Institutionalisation contributes non-questionable expectations of either party and the existence of contact patterns. The other aspect of long-term relationships is the adaptations. Adaptations refer to changes or investments accomplished by the parties involved in the specific relationship to facilitate the specific client–supplier collaboration (Van der Valk et al. 2008).

The *interacting parties* include the client firm with its actors and the supplier firm with its actors. A number of characteristics of the parties influence the interaction process. Firstly, technology determines not only how parties might interact but also defines the product and manufacturing process of both parties, which in effect ties the two parties together. Secondly, organisational size, structure and strategy determine the power the involved parties have during interactions. For example, a large firm has a greater possibility to dominate interactions in comparison with a small firm. The firm's experience of similar relationships will contribute knowledge about the management of the specific relationship. Experiences also influence the individuals perceived level of importance of a relationship and, in turn, the firm's commitment to that relationship (Håkansson 1982).

The interaction process is influenced by two contextual dimensions: the environment and the atmosphere. The *environment* is defined in terms of for example market dynamism, market structure and internalisation of the market. Environmental aspects influence among all the parties' ability to anticipate and predict changes in the market and firms' inspiration to join international relationships. A great geographical distance between client and supplier brings cultural and language barriers which in turn are challenges for relationship management. The *atmosphere* can be described in terms of dependence, power, conflict, cooperation and trust. The variables of the atmosphere are often intertwined, which is a result of the specific relationship (Håkansson 1982): the power-dependence relationship among the involved firms, the state of conflict or cooperation, overall closeness or distance of the relationship and firms' mutual expectations and trust. Specific exchange episodes and long-term exchange experiences contribute to the evolvement of the atmosphere.

3.2 Limitations of the Interaction Approach

The interaction approach has been applied by Kern and Willcocks (2002) for exploring IS outsourcing relationships. Through their research, they found that this approach has both potentials and limitations for explaining IS outsourcing relationships. The limitations identified are related to that the interaction approach does not:

- differentiate among the diverse features of the client and the supplier
- include the significance of the contract and the management of client–supplier relationships

The limitation of not including the features of the parties involved in the interaction process can partly be explained as a result of that the individual dimension in the interaction approach does not explicitly regard the numerous actors that participate in business interactions (Axelsson 2010). Bringing the right people into the interaction process is not only about the right position, role and competence, but also about building a relation with the counterpart. The activities performed by individuals are often studied as activities inside and in-between firms, not so much as actions by individuals. The limitation is addressed in this research study by viewing IS outsourcing relationships as constituted of:

- actors working at the business, the process and the IS level of the client firm and the IS supplier firm and
- interactions within and between actors at the client firm and the IS supplier firm.

The idea of viewing the client firm and the IS supplier firm as constituted of three levels is adopted from Österle (1995) and Tolis and Nilsson (1996). To achieve successful results, every organisational change should incorporate the business strategy, the processes and the IS of the firm. Firms that continually focus and improve the interactions among actors of these three levels establish a bound which contributes mutual understanding and acceptance of organisational changes, such as outsourcing of IS activities.

The interaction approach should further emphasis the contract and the management of interorganisational relationships. The contract in IS outsourcing is perceived as the foundation for the relationship and needs therefore careful consideration (Kern 1999). It is found that during the execution of IS outsourcing the management is largely focused on managing relations (Kern and Willcocks 2002). The reported limitations are addressed in this research study through the inclusion of the importance of the contract and the management of IS outsourcing relationships.

To summarize, the following improvements should be incorporated in the interaction approach to increase the description of IS outsourcing relationship interactions:

- the client firm and the IS supplier firm as constituted of three levels; business level, process level and IS level and
- the dimensions contract and management.

4 Conditions Influencing Client–IS Supplier Interactions

The conditions influencing client–IS supplier interactions during IS outsourcing are presented in the form of a conceptual framework. According to Miles and Huberman (1994), a conceptual framework explains the main things to be studied through descriptive and graphical illustrations. This includes the key conditions, constructs or variables and the presumed relationships among them. The conceptual framework is (often) derived from general objectives in theory and prior empirical research. The conceptual framework in Fig. 2 describes the conditions influencing IS outsourcing relationship interactions by complementing the interaction approach with the presented improvements (hereafter referred to as elements).

The added elements are formatted in bold in Fig. 2 to highlight the changes compared with the original model of the interaction approach (see Fig. 1). The client firm and the IS supplier firm are both viewed as being comprised of three levels. The interaction process constitutes the inter-firm interactions among actors working at business, process and IS level (see further Fig. 3). The position of the dimensions contract and management below the arrow of the interaction process is a result of the significance of the contract and the management to reach long-term relationships (Cullen et al. 2005). The dimensions concern client–IS supplier relationships and are therefore positioned between the client and the IS supplier. The dimensions are presented together with conditions identified as influencing IS outsourcing relationship interactions. In conclusion, the conceptual framework and its elements are not organised according to a certain sequential order but are instead intended to emphasis the conditions related to the interaction process of IS outsourcing relationships. The added elements of the interaction approach are further described below.

4.1 Firm Levels and Intra- and Inter-firm Interactions

During the execution of IS outsourcing different actors are involved. The actors are situated at three different levels of the client firm and the IS supplier firm. The two-way arrows in Fig. 3 illustrate the possible intra-firm and inter-firm interactions in IS

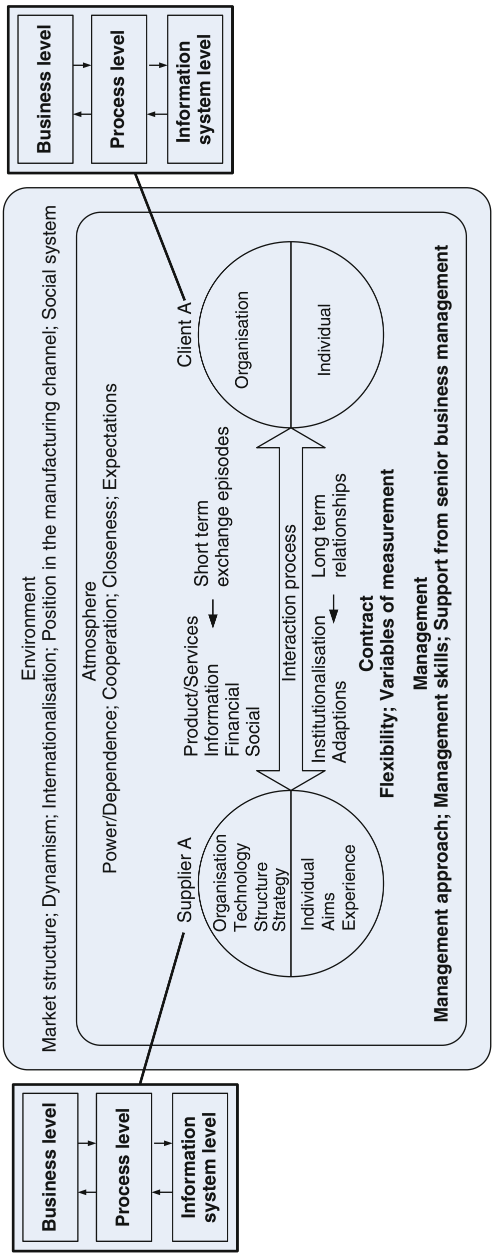


Fig. 2 A conceptual framework for describing conditions influencing IS outsourcing relationship interactions. (The author's; adapted from Håkansson (1982))

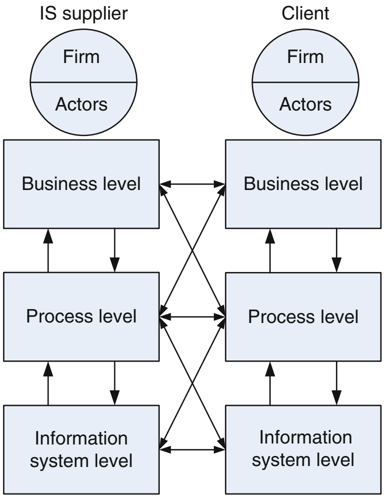


Fig. 3 Firm levels, actors and actors' possible interactions. (Adaptation of Österle 1995, p. 30)

outsourcing relationships. The *business level* of the firm concerns the firm's position in the market and the firm's key decisions (Österle 1995). Business development focuses on the firm's relations with different actors such as clients, suppliers and other business partners, in the market environment. The aim of business development is to focus on the business strategy to improve business relations. The *process level* concerns the implementation of the firm's business strategy by specifying the firm structure and deriving outputs, flows, computer support and management tools (Österle 1995). Business process development is about designing new and better workflow among different business processes in the firm. The *IS level* concerns the implementation of the business process design. This is partly conducted by delivering guidelines that emphasis information, data and IS (Österle 1995). Development of IS focuses on efficient allocation of resources and enablers to achieve professional business processes.

The importance of including different firm levels is particularly due to the impact IS outsourcing has on the firm and its performance (Feeny and Willcocks 1998). The impact is partly a result of that IS activities are intertwined in more or less all business processes. The result of including actors representing different firm levels is a higher acceptance of IS outsourcing. For example is it found that IS outsourcing success is more likely to be reached when interactions between senior business managers and senior IT managers guide the IS outsourcing decision (Lacity and Willcocks 1998).

4.2 The Contract Dimension

The contract is often expressed as the foundation of IS outsourcing relationships. This is partly a result of that the contract provides a bound that specifies each actor's rights, duties and responsibilities. The goals, policies and strategies underlying IS outsourcing are articulated in the contract. The aim of the contract is to facilitate exchange and cooperation to reach better performance.

To reach a successful contract it is recommended that it should be as descriptive as possible (Fisher et al. 2008). The description of the contract should include clauses that refer to evolution, reversibility, termination and penalisation. Further, the *flexibility* of the contract is important in order to reflect evolution of technology, firm needs, relationship development and the emergence of new competitive services. To be able to judge if expectations are met, *variables of measurement* are significant to include in the contract (Kumar and Palvia 2002). However, it is difficult to define services in quantifiable terms. For example, the quality of a system design report, user-friendliness across applications and the maintainability of the IS are all examples of IS outputs that are difficult to measure quantitatively. A well-developed contract influences the efficacy of IS outsourcing relationships and is a critical condition of success (Alborz et al. 2003).

The contract in itself, however, is not a guarantee for achieving successful IS outsourcing relationships. To reach successful relationships a relatively intimate relationship should be established between client and IS supplier (Kern and Willcocks 2001). The existence of such client–IS supplier relationships reduces the need for detailed monitoring of the performance of each party. Long-term relationships and on-going communication promote the institutionalisation of relationships, which in turn brings a state of condition recognized by that clients and IS suppliers are comfortable with each other. Besides a thoroughly developed and well described IS outsourcing contract, the establishment of intimate and cooperative long-term IS outsourcing relationships is crucial for reaching successful outcomes of IS outsourcing.

4.3 The Management Dimension

The management before and after the contract has been signed has great influence on IS outsourcing outcomes. Relationship management is a complex task and to be successful managers need to be aware of the environment in which the relationship exists. Other relations and internalisation of the market are part of this environment. During IS outsourcing ineffective management brings the risks for loss of control, high contract management costs and backsourcing (Cullen et al. 2005). The importance of IS outsourcing management can be explained by, for example, the complexity of IS and the rapid evolution of the IT field.

It is found that post-contract management is largely spent on managing relations (Kern and Willcocks 2002). Three conditions related to post-contract management are *management approach, management skills* and *support from senior business management*. A shared management approach contributes to successful performance of problem solving, similar values and favourable personal chemistry among actors (McFarlan and Nolan 1995). Management skills required during the IS outsourcing process are especially related to communication (Alborz et al. 2003). Communication is a significant ‘tool’ that makes it possible to influence, and most importantly, negotiate across firm boundaries. Communication is crucial to reach mutual understanding about how IS outsourcing relationships should be designed and activities performed during IS outsourcing. Communication is moreover found to be important in one way or another when performing IS activities. The less structured the outsourced IS activity is the more effort needs to be put on managing IS outsourcing (Wüllenweber et al. 2008). Further, support from senior business management is important during IS outsourcing management (Kern and Willcocks 2001). The support should be expressed through the initiation of change, explanation of reasons for IS outsourcing and the promotion of appropriate management and resources. The management needs to be adjusted to suit the requirements of the actual case of IS outsourcing. To meet these requirements, the engagement of both client and IS supplier plays a significant role.

5 Conclusions

Outsourcing of IS activities is becoming a trend in many firms. Client–IS supplier relationships arising as a result of outsourcing bring challenges that need to be understood and managed. Therefore managers should be aware of the features of this relationship so that the probability of IS outsourcing to become successful is higher. Research should enhance the knowledge of IS outsourcing relationships for reaching IS outsourcing success. This paper aimed to fill the gap by describing conditions influencing client–IS supplier interactions during IS outsourcing.

The theoretical contribution of this research study is an adequate conceptual framework for studying interorganisational relationships; particularly IS outsourcing relationships and their interactions. The conceptual framework presented in Fig. 2 is a further development of the interaction approach and emphasises conditions related to IS outsourcing relationships.

From a practical perspective the conceptual framework is intended to be used by managers at different levels of the client firm and the IS supplier firm. It can be used for purposes such as the following:

- forming the basis of managerial decisions regarding client–IS supplier relationships,
- focusing the attention of actors on conditions influencing the interaction process of IS outsourcing relationships and
- facilitating for firms to realize IS outsourcing, make decisions and perform actions related to IS outsourcing relationships.

Conceptual frameworks that address outsourcing options are required to better understand differences among options and to find out what makes IS outsourcing more or less successful (King and Torkzadeh 2008). The conceptual framework presented in this paper contributes to the literature on IS outsourcing relationships and highlights the conditions influencing IS outsourcing relationship interactions. When adopted the conceptual framework can contribute to a better understanding of challenges of IS outsourcing relationships and further provide guidance for IS outsourcing relationship management. To increase the practical usefulness of the conceptual framework it should be tested empirically. An empirical study may contribute information about significant client–IS supplier interactions and the recognition of key conditions for reaching successful IS outsourcing relationships. Further, an empirical study may contribute with information about which actors that are best suited to apply the conceptual framework.

The interaction approach has been criticized for not putting enough attention on the external context surrounding the specific client–IS supplier relationship (Axelsson 2010). In IS outsourcing this aspect becomes important since the actors will be working in different geographical locations. One criticism builds upon that interactions aren't just part of a dyadic process but rather are connected in a network-like process (Ford et al. 2010). Firms interact with several other firms and interactions between two firms may affect their interactions with these others. Dyadic relationships are thus conditioned by relationships outside the dyad and should thereby be looked upon as embedded in a network of connected relationships. For further research it should therefore be interesting to study IS outsourcing relationships with regard to a relationship network perspective.

References

Alborz S, Seddon PB, Scheepers R (2003) A model for studying IT outsourcing relationships. In: Proceedings of the 7th Pacific Asia conference on information systems, 10–13 July, Adelaide, South Australia, pp 1297–1313

Alsudairi M, Dwivedi YK (2010) A multi-disciplinary profile of IS/IT outsourcing research. J Enterp Inf Manage 23(2):215–258

Axelsson B (2010) Business relationships and networks: reflections on the IMP tradition. IMP J 4(1):3–30

Bhatt P, Shroff G, Anantaram C, Misra A (2006) An influence model for factors in outsourced software maintenance. J Softw Maint Evol Res Pract 18(6):385–423

Cullen S, Seddon PB, Willcocks L (2005) Managing outsourcing: the lifecycle imperative. MIS Q Exec 4(1):229–246

Dabhilkar M, Bengtsson L, von Haartman R, Åhlström P (2009) Supplier selection or collaboration? Determining factors of performance improvement when outsourcing manufacturing. J Purch Supply Manage 15(3):143–153

Ekman P (2006) Enterprise systems & business relationships. The utilization of IT in the business with customers and suppliers. PhD thesis, Mälardalen University School of Business, Västerås, No 29

Feeny D, Willcocks LP (1998) Core IS capabilities for exploiting information technology. Sloan Manage Rev 39(3):9–21

Fisher J, Hirschheim R, Jacobs R (2008) Understanding the outsourcing learning curve: a longitudinal analysis of a large Australian company. Inf Syst Front 10(2):165–178

Ford D, Gadde L-E, Håkansson H, Waluszewski A (2010) Analysing business interaction. IMP J 4(1):82–103

Goles T, Chin WW (2005) Information systems outsourcing relationship factors: detailed conceptualization and initial evidence. DATA BASE Adv Inf Syst 36(4):47–67

Gonzales R, Gasco J, Llopis J (2006) Information systems outsourcing: a literature analysis. Inf Manage 43(7):821–834

Håkansson H (ed) (1982) International marketing and purchasing of industrial goods: an interaction approach. Wiley, Chichester

Kern T (1999) Relationships in information technology outsourcing: an exploratory research study of a conceptual framework. Unpublished PhD thesis, University of Oxford, Christ Church, Oxford, No 468

Kern T, Willcocks L (2000) Exploring information technology outsourcing relationships: theory and practice. J Strategic Inf Syst 9(4):321–350

Kern T, Willcocks L (2001) The relationship advantage: information technologies, sourcing and management. Oxford University Press, Oxford

Kern T, Willcocks L (2002) Exploring relationships in IT outsourcing: the interaction approach. Eur J Inf Syst 11(1):3–19

King WR, Torkzadeh G (2008) Information systems offshoring: research status and issues. MIS Q 32(2):205–225

Kumar N, Palvia PC (2002) A framework for global IT outsourcing management: key influence factors and strategies. J Inf Technol Cases Appl 4(1):56–75

Lacity M, Willcocks L (1998) An empirical investigation of information technology sourcing practices: lessons from experience. MIS Q 22(3):363–408

Lacity M, Willcocks L (2000) Relationships in IT outsourcing: a stakeholder perspective. In: Zmud RW (ed) Fraiming the domains of IT management—projecting the future through the past. Pinnaflex Education Resources, Cincinnati, pp 355–384

Leek S, Turnbull P, Naude P (2000) Is the interaction approach of any relevance in an IT/E-commerce driven world? In: Proceedings of the 16th industrial marketing and purchasing conference (IMP 2000), University of Bath, UK. www.impgroup.org. Accessed 1 March 2008

McFarlan FW, Nolan RL (1995) How to manage an information technology outsourcing alliance. Sloan Manage Rev 36(2):8–23

Miles MB, Huberman AM (1994) Qualitative data analysis. Sage, California

Österle H (1995) Business in the information age—heading for new processes. Springer, Berlin

Tolis C, Nilsson AG (1996) Using business models in process orientation. In: Lundeberg M, Sundgren B (eds) Advancing your business: people and information systems in concert, vol VIII. Studentlitteratur, Lund, pp 1–24

Van Der Valk W, Wynstra F, Axelsson B (2008) An empirical investigation of interaction processes between buyers and sellers of business services. IMP J 2(2):3–24

Wüllenweber K, Beimborn D, Weitzel T, König W (2008) The impact of process standardization on business process outsourcing success. Inf Syst Front 10(2):211–224

Facilitating the Creation of Quality Object Models Using a Knowledge Based System

Narasimha Bolloju and Christoph Schneider

Abstract In today's systems development environments, object models are becoming increasingly important. However, given the complexity of UML, it is difficult to create quality object models; further, while current CASE tools can aid in drawing object models, they do not provide much support for enhancing the quality of object models. To this end, we developed a prototype of a knowledge-based system designed to facilitate the creation of quality object models by novice analysts. Using a design science approach, we provide a description of the design objectives, the system architecture and implementation details, and discuss results of an initial evaluation of the systems' efficacy. The analysis of the system demonstrates its ability to reliably extract relevant information from use case descriptions, and can lead to significantly more correct classes, attributes, and relationships being identified; however, the analysis has shown that the tool has not prevented novice analysts from adding incorrect elements to object models.

1 Introduction

Today's tightly connected world has seen rapid advances of information systems used by individuals and business organizations. This progress has fuelled rapid changes in information systems development (ISD) methodologies, leading to new and continuing challenges for information systems professionals. In ISD, use case models are widely used for capturing information system requirements in terms of use cases diagrams and descriptions, which are later used to develop object models by identifying classes with attributes and operations and relationships among those classes. With the increasing popularity of UML and object oriented programming languages, object modelling became an important step in the systems development process, and object models are considered extremely useful in supporting activities such as client verification, clarifying technical understanding, programmer specifications

N. Bolloju (✉)
Department of Information Systems, City University of Hong Kong, Hong Kong, China
e-mail: narsi.bolloju@cityu.edu.hk

J. Pokorny et al. (eds.), *Information Systems Development,* 137
DOI 10.1007/978-1-4419-9790-6_11, © Springer Science+Business Media, LLC 2011

and maintenance documentation. Consequently, object modelling skills are a crucial component of the skill set of today's information systems professionals.

However, acquiring these skills is often challenging, especially for beginners, as the tasks and solutions are ill-defined [see 4, 14, 21, 22]. Further a key challenge in developing object models is the usability and complexity of UML, which has been demonstrated to be two to eleven times more complex than other modelling methods [21], and even some experienced analysts may not be using this method due to a lack of understanding [6]. Whereas computer-aided software engineering (CASE) tools aid in drawing, documenting, and navigating through object models [7], these tools typically lack features that support high-quality object modelling.

Our current study attempts to help address this shortcoming by focusing on the main research question: *how can CASE tools be enhanced to support object modelling activities performed by novice systems analysts?* In a first attempt to answer this question, we propose features to enhance the completeness and validity of object models. Specifically, following a design science approach [10], we propose and build a knowledge-based system that analyzes the completeness and validity of object models and provides recommendations for improving these aspects of the model. In the following section, we provide an overview of our research method, including a brief discussion of design science and its applicability to our problem. Then, we will present an overview of the key contribution of this project, namely, a detailed description of the tool. This tool is tested using student subjects in order to assess its efficacy in improving the quality of object models created. Then, we will present the results of this preliminary assessment, and conclude with a discussion of our findings.

2 Design Science Research

A key goal of this research is to facilitate the creation of high quality object models, and in particular, UML diagrams, so as to further their use by novice and experienced analysts. Hence, a key outcome of this study is the creation of an artifact that provides an initial step towards achieving this goal. This research largely follows a design science approach; as opposed to natural sciences, which attempt to gain a deeper understanding of phenomena, design science focuses on developing tangible artifacts that serve a designated purpose, and the outcomes are evaluated based on the utility [13]. Considering business needs and theoretical foundations for the development and evaluation of the artifact ensures both relevance and rigor of our research [10]. In our research process, we followed a multi-step approach [see 12, 15] consisting of problem identification, definition of design objectives, the design process, and evaluation. In the next section, we will delve deeper into the problem to be addressed by the prototype.

3 Challenges and Sources of Support for Object Modelling

When developing conceptual models, novice or inexperienced systems analysts encounter difficulties in the areas of domain-specific knowledge, problem-structuring, and the cognitive processes associated with identifying model elements [19]. Further, lack of understanding of the domain by requirement engineers or systems analysts and miscommunication between the users and technical persons are two common problems associated with early stages of systems development [24]. Hence, the effectiveness of a conceptual model is affected by the complex relationships that exist between modelling constructs, the task requirements, the analyst's modelling experience and cognitive abilities, and the interpreter's prior exposure to conceptual models [25]. The absence of standardized validation procedures makes conceptual modelling a complex task for novice analysts to perform efficiently and effectively [20].

In order to address such problems, several sources for support exist. These facilities supporting the development of object models can be categorized as generation focused, guidance focused, or critique focused approaches. Generation focused approaches use natural language processing and expert system technologies to attempt to automatically generate object models from textual descriptions of a system [e.g., 11, 16, 26]. Guidance focused approaches do not attempt to create models, but rather guide the analyst through the process; these approaches typically use mechanisms such as practitioner-oriented recommendations for developing quality models (in the form of guidelines for various tasks such as identifying classes and relationships, naming and presentation conventions, and use of analysis patterns, see, e.g., [3]), analysis patterns and frameworks for specific domains [e.g., 5, 8], and wizards [e.g., 1]. In contrast, critique focused approaches provide advice on the analyst's actual models during or after the creation of the models. For example, ArgoUML [17, 18] uses agents that continuously check an existing model for potential problems and advise on areas of improvement so as to address the cognitive needs of software designers.

Each of the above approaches has certain strengths and limitations. Obviously, generation focused approaches appear to be attractive due to automation, but depend heavily on the quality of inputs and do not scale up very well to real life systems. Guidance focused approaches provide exhaustive lists of dos and don'ts, but are not built into CASE tools; thus, only large numbers of generic guidelines, rather than specific advice, are provided. Similarly, critique focused approaches, which are normally expected to identify errors in a given model and provide recommendations, mostly offer broad sets of recommendations in the form of lengthy checklists that are often drawn from guidance based approaches. They do, however, offer learning opportunities for inexperienced analysts. Overall, neither of these sources of support can address the specific challenges (especially novice) analysts face.

4 Design Objectives and Process

To overcome the challenges novice analysts face when developing object models, our objective was to design a prototype of a knowledge-based system (KBS) that can build on the strengths of the different approaches. Specifically, we envisioned a system that can make use of a knowledge base to provide guidance in the form of specific recommendations that are based on the system requirements and the object model created.

A survey of students—administered after completing a course on systems analysis and design—asking to list the three major challenges in object modelling revealed that identifying required classes, required attributes for a given class, relationships, and required operations were seen as the most difficult tasks. Hence, we focused on these aspects in the design for our prototype. In order to build a system that can both be used to enhance object models and facilitate the learning of novice analysts, we sought to design a domain-independent modelling support system that helps novice analysts in ensuring that the object models capture the required elements. Our design process involved several iterations of building and testing the prototype. The initial prototype was used to generate and test a knowledge base that incorporates common collaboration patterns (e.g., most *role* classes are associated with *transaction* classes). After two major iterations (and multiple minor iterations) of refining and testing the system, we implemented the current functionality, namely, to build a domain-independent system. Key to building a domain-independent system is to extract the pertinent information from a system's specifications. Given the widespread use of use cases for preparing object models, we extended the initial prototype so as to automatically extract the relevant information from use cases; based on the information contained in the use cases and an initial object model, recommendations are provided for improving the quality of the object model. In the next section, we will provide an overview of the system and discuss details related to the implementation of the functionality.

5 System Overview and Implementation

The prototype system was developed as an extension to ArgoUML, an open-source CASE tool [23] that closely follows the UML standard. The prototype provides a function for invoking a knowledge-based system component to obtain recommendations for improvements to the current version of an object model created using ArgoUML functionality. For this purpose, it is assumed that use cases are specified first to describe the expected functionality of the system under consideration.

Typical usage of the system requires the user to start a project in ArgoUML with one use case diagram and provide descriptions of the use cases present in that diagram in terms of trigger, pre- and post-conditions, and descriptions of the steps in the main and alternate scenarios. The user may then create an initial draft of an object model, either in parallel to or after the use case specification. Upon invocation of the object model analyzer function, the system extracts relevant information from the use cases using natural language processing (NLP) techniques and determines possible pattern player types for each class in the object model. Then, it makes inferences to identify missing and invalid elements in the draft object model, and finally generates recommendations for improving the model. The user is expected to iterate through the activities of reviewing the recommendations, applying any relevant recommendations to the object model and invoking the object model analyzer function as long as some applicable recommendations are found. It is possible that some of the recommendations related to invalid elements (e.g., an attribute present in the model but not included in any use case) may also require modification of use cases.

The KBS, implemented using several open source software tools and libraries, was mostly developed using Java 1.6 and Jess [9]. ArgoUML [2] was selected because it is written in Java and it can be easily integrated with any external Java libraries. ArgoUML provides a set of application programming interfaces to extend its functionality in the form of extension modules. Further, we used various other open source tools, such as Protégé (ontology editor), Jess (inferencing engine), and JessTab (to integrate Protégé and Jess) for this prototype implementation.

The KBS system consists of four primary components: (a) use case description support, (b) adapter, (c) knowledge base, and (d) processors. The first component offers an interface for adding descriptions for the use cases present in a use case diagram. The adapter component facilitates the interaction between ArgoUML and components of the KBS. This adapter translates object models in ArgoUML data structures into ontology instances and controls the interaction between ArgoUML and other system components. The knowledge base component maintains collaboration pattern knowledge and diagnostic rules, which are expressed as a combination of an ontology and a set of Jess rules. OWL ontology was used to represent use case descriptions, the object model under development, and the collaboration patterns.

The processor components perform the following activities when the user invokes the system functionality through ArgoUML: (a) convert use cases and the current version of the class diagram into OWL ontology, (b) analyze use cases and create an ontological representation of actions and recognized entities, (c) optionally, collect inputs related to archetype assignments (*People*, *Events*, *Places*, *Things*) to recognized elements in use cases and class diagram, (d) transform the current ontology into facts in the format required by Jess, (e) perform inference for completeness and validity by applying heuristics on the facts loaded from the Jess file, (f) report suggestions regarding missing and invalid classes, attributes and relationships, and (g) save the ontology as an XML file and Jess facts as a text file for subsequent use.

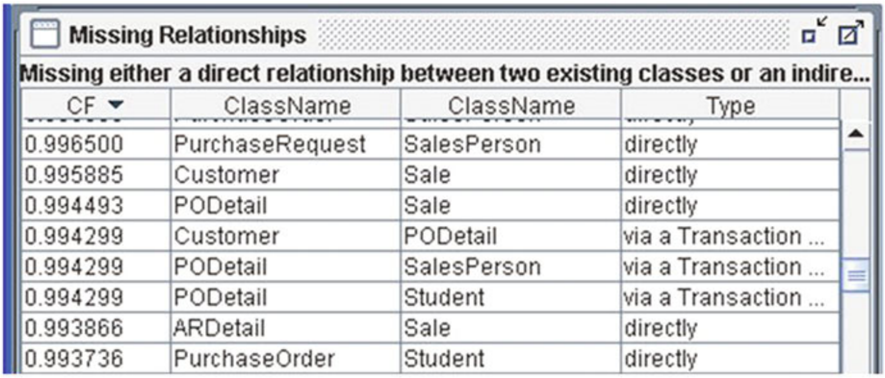
As can be seen from this description, two key activities of the KBS are to analyze the use cases and the diagnostic process that involves applying heuristics. The Use Case Analyzer uses natural language processing techniques for analyzing a given set of use cases and transforming the analyzed information into an intermediate form that works as basis for both object model quality analysis and glossary management. The intermediate form is an OWL ontology that describes the relationships between subjects, verbs, and objects in the use cases. The Use Case Analyzer processes use cases by analyzing each scenario step of each scenario. For each step, it generates a sub-network of the ontology. Having translated the ontology into facts, the KBS proceeds with diagnosing the object model.

This diagnostic step involves heuristics that are primarily derived from practitioner guidelines, as well as from data modelling and object modelling processes employed by experts (e.g., commonly occurring patterns). The following is an example of a heuristic to illustrate the diagnostic processes related to identifying missing relationships.

There is a missing relationship between a pair of distinct classes c1 and c2
IF there does not exist a relationship that connects c1 and c2
AND
there exist two pattern players p1 and p2 derived for c1 and c2 respectively
such that <p1,p2> is a valid collaboration pattern
AND
there exist two recognized entities e1 and e2 from the use cases
such that e1 maps to c1 and e2 maps to c2 and
e1 and e2 belong to the same use case description

This heuristic determines whether a relationship is required between any given pair of classes in the object model considering the pattern type of those classes (e.g., *role* and *transaction*) and a corresponding collaboration pattern (e.g., most *role* classes are associated with *transaction* classes) provided any entities of similar types are present in a single use case. For example, if the object model has two classes *Agent* and *Sale,* an association is required if some use case has *Agent* as an actor and has certain steps referring to *Sale*.

A sample of recommendations provided for missing relationships is shown in Fig. 1. Each row points to a possible missing relationship between a pair of existing classes. The suggestion also includes details such as whether a direct or an indirect relationship is required and the confidence in that recommendation represented by a confidence factor (CF) between 0 and 1.


Missing Relationships

Missing either a direct relationship between two existing classes or an indire...

CF	ClassName	ClassName	Type
0.996500	PurchaseRequest	SalesPerson	directly
0.995885	Customer	Sale	directly
0.994493	PODetail	Sale	directly
0.994299	Customer	PODetail	via a Transaction ...
0.994299	PODetail	SalesPerson	via a Transaction ...
0.994299	PODetail	Student	via a Transaction ...
0.993866	ARDetail	Sale	directly
0.993736	PurchaseOrder	Student	directly

Fig. 1 An example of recommendations provided

6 Evaluation of the Knowledge-Based System

Our initial test of the system involved testing the system's ability to analyze use cases and apply the diagnostic heurists. Several sets of use cases and draft domain models belonging to different business application domains were used for this testing. Based on these tests, we have refined the use case analyzer component and the diagnostic rules. Significant changes to the user interface components (e.g., presentation of diagnostic results as tables rather than text) were also incorporated following the feedback obtained from the testers.

Having established that the system functioned as expected, we proceeded to conduct another, more formal evaluation. Specifically, we conducted a controlled study using undergraduate students enrolled in a course on systems analysis in order to evaluate the improvements of object models created.

A total of 84 undergraduate students of information systems in the college of business of a large metropolitan university in Far East Asia participated in the study. These students were enrolled in three sections of a systems analysis course that mainly covers the object-oriented approach with selected UML techniques. The study was conducted toward the end of a 13-week course; by that time, the students had become reasonably familiar with object modelling while working on a running case study in the laboratory sessions and another case study in their project work. The study was conducted as an exercise during a 100-min laboratory session and the exercise contributed 5% to the students' grade. Each laboratory session included the following activities, in addition to a questionnaire on prior knowledge and skills related to object modelling:

1. The instructor provided a handout of general instructions about the experiment and explained the procedures (10 min)
2. Each subject developed an object model (version 1 class diagram) for the "Small Bytes" subscription system (see Appendix) using ArgoUML (35 min)
3. Break (10 min)
4. Each subject received feedback on the class diagram from the Knowledge Based System. Based on this feedback, the subjects were given time to revise their initial version of the object model (35 min)

We then assessed the quality of the initial and the revised object models in order to determine the improvements to the model based on the recommendations from the system. Considering the fact that object models created (draft and final versions) could have several variations and the number of object models to be assessed, we employed a two-stage assessment process. The first stage involved automated assessment by matching classes, attributes and relationships of a given class diagram against the model solution. The second stage involved manually resolving any unmatched elements of the class diagrams. Thus, the second stage supplemented the automated assessment performed in the first stage by providing opportunities for validating alternative representations of object models for the given problem domain.

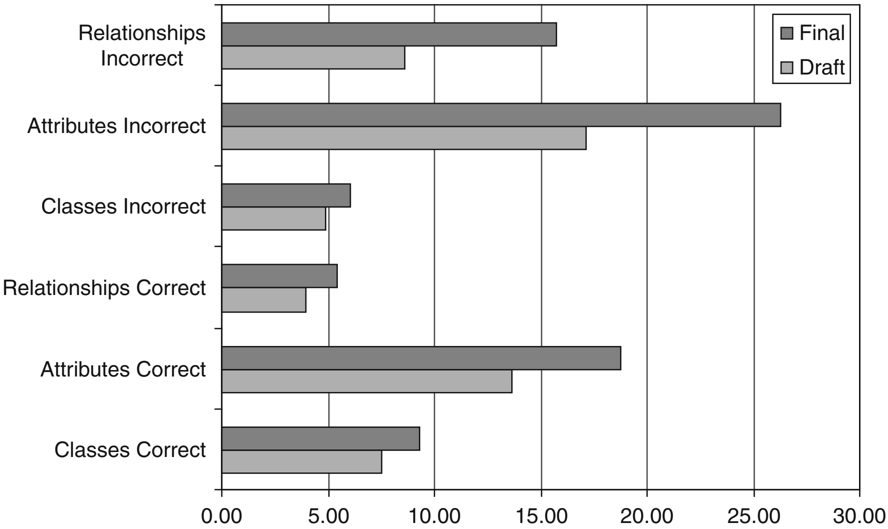


Fig. 2 Average number of model elements in draft and final versions

Since the first assessment stage matched about 70% of the model elements, the second assessment stage has been quite efficient and minimized subjectivity (based on a good inter-rater agreement).

We tested the changes in model quality using repeated measures ANOVA in SPSS 15.0. Figure 2 depicts the preliminary findings obtained from the analysis of the correct and incorrect elements in the draft and final versions of object models created by the subjects. While, as expected, the final versions contained relatively more correct elements compared to draft versions (with significant ($p < 0.001$) quality improvement in classes attributes, and relationships identified), the average numbers of incorrect or wrong elements have also increased significantly.

7 Discussion and Conclusion

This paper has provided an overview of the development of a knowledge-based system to aid systems analysts in creating higher quality object models. Following a design science approach, we have created an extension to the ArgoUML open-source CASE tool; using natural language processing, the KBS analyzes use case descriptions and, based on heuristics, provides recommendations to improve object models. An analysis has shown that the KBS can reliably extract information from use cases, helping novice analysts to identify required elements in object models. Our prototype can thus be seen as a proof of concept, demonstrating the efficacy of using natural language processing and inferencing in the domain of use cases. However, our analysis has also shown that the KBS could not prevent the subjects from adding incorrect elements, nor did it reduce the number of incorrect elements. This increase may be due to subjects not following the recommendations or adding yet other elements; a more fine-grained qualitative assessment of the object models may shed more light on this interesting finding. Future research should attempt to improve the tool in this regard, or attempt to find ways to ensure that the analysts carefully consider the tool's recommendations. A potential way to achieve this would be to guide the analyst through a series of iterations, so as to help the analyst focus on the most important errors and omissions of the diagram.

A limitation of this study may be the use of students working on a relatively small problem. However, the development of our tool was targeted at novice analysts, so our subject pool appears adequate. The current test of the prototype was geared at demonstrating the functionality of the system, as well as the ability to provide guidelines that help to improve object models; thus, we have not directly tested the KBS against other alternative tools, but regard this as a good direction for future research.

In sum, we see this prototype as a first step in overcoming limitations of current guidance and critique focused approaches to enhancing the quality of object models. The prototype of the KBS has shown that automated tools can provide valuable guidelines to improve object models; further, the ability to analyze use case descriptions enables the tool to be used across different domains, yet give recommendations specific to the problem at hand. Hence, we see this tool as very valuable in business contexts. Especially for small enterprises, who may not be able to afford large teams of analysts (and thus cannot draw on dedicated resources for providing critique and guidance), such tool could prove valuable for creating quality object models. Another potential application would be the use of the KBS in a learning context, especially for novice analysts. Specifically, the tool can be used to train novice analysts in identifying relevant elements from given use case descriptions, thus helping to improve their skills in generating initial versions of object models. While we have demonstrated that the KBS can be used to improve object models, future research should explore the effects of the guidance provided on learning, especially in comparison to generic guidelines or other approaches.

Acknowledgements The work described in this paper was substantially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CityU 110207).

8 Appendix: Small Bytes Subscription System Case Study

A small independent software journal, Small Bytes, has asked you to design a new system for managing its subscriptions and editorials. While the concept of managing subscriptions is quite straightforward, the details are numerous, as described below:

Small Bytes is published on a monthly basis; a typical monthly issue consists of 5–10 articles, each written by one or more authors in the software engineering field.

Though the authors receive no payment for their articles, they do receive a year's free subscription as a token of appreciation for their efforts; if they already have a subscription, then the expiration date is extended for a year.

Small Bytes has an editorial board of advisors, some of whom may also be authors from time to time; each advisor on the editorial board normally serves for a one-year or two-year term, and receives a complimentary subscription to the magazine for that term. Three or four members of the editorial board, selected by the editor, review each submitted articles and provide their comments to the editor. As with most magazines, issues are scheduled and planned months in advance.

Small Bytes is managed by an editor supported by an assistant who is responsible for keeping track of subscriptions and submissions using an in-house computerized system. The assistant receives subscription orders (new and renewal) and new article submissions via mail or fax, and registers those orders and submissions into the system. The editor does a quick review of the submissions to ensure that they fit the magazine before assigning three or four advisors for a detailed review and comments. Upon receiving replies from the advisors, the assistant records the review comments into the system. From time to time, upon receiving instructions from the editor, the assistant updates the list of advisors and their subject areas.

Small Bytes is sold on a subscription basis; most subscriptions are for a one-year period, but the publisher accepts subscriptions for periods longer than or shorter than a year by simply pro-rating the annual subscription price. Payments for new subscriptions are normally received by cheque. Some subscribers pay by credit card, but the publisher insists (because its bank insists) that credit card payments be accompanied by a signature; this means that the credit card orders are typically sent by fax or mail. As noted earlier, contributing authors and members of the editorial board of advisors receive a complimentary one-year subscription to the magazine.

References

1. Antony SR, Batra D (2002) CODASYS: a consulting tool for novice database designers. ACM Sigmis Database 33(3):54–68
2. ArgoUML (2006) ArgoUML version 0.20. http://argouml.tigris.org/. Retrieved 6 Aug 2007
3. Bolloju N (2004) Improving the quality of business object models using collaboration patterns. Commun ACM 47(7):81–86
4. Borstler J, Hadar I (2008) Pedagogies and tools for the teaching and learning of object oriented concepts. Lect Notes Comput Sci 4906:182
5. Coad P, North D, Mayfield M (1997) Object models: strategies, patterns, and applications, 2nd ed. Prentice-Hall, Upper Saddle River
6. Dobing B, Parsons J (2006) How UML is used. Commun ACM 49(5):109–113
7. Eriksson HE, Penker M, Lyons B, Fado D (2004) UML 2 toolkit. Wiley, Indianapolis
8. Fowler M (1997) Analysis patterns: reusable object models. Addison-Wesley, Reading
9. Friedman-Hill E (2005) Jess version 7.0b6. http://eprints.agentlink.org/5380/. Retrieved 19 April 2010
10. Hevner AR, March ST, Park J, Ram S (2004) Design science in information systems research. MIS Q 28(1):75–105

11. Kaindl H (2004) Active tool support for requirements engineering through RETH. In: Proceedings of the 12th IEEE international requirements engineering conference, Kyoto, Japan, pp 362–363
12. Kuechler B, Vaishnavi V (2008) On theory development in design science research: anatomy of a research project. Eur J Inf Syst 17(5):489–504
13. March ST, Smith GF (1995) Design and natural science research on information technology. Decis Support Syst 15(4):251–266
14. Moritz SH, Blank GD (2005) A design-first curriculum for teaching Java in a CS1 course. ACM SIGCSE Bull 37(2):89–93
15. Peffers K, Tuunanen T, Rothenberger MA, Chatterjee S (2007) A design science research methodology for information systems research. J Manage Inf Syst 24(3):45–77
16. Purao S (1998) APSARA: a tool to automate system design via intelligent pattern retrieval and synthesis. ACM SIGMIS Database 29(4):45–57
17. Robbins JE, Redmiles DF (1998) Software architecture critics in the Argo design environment. Knowl Based Syst 11(1):47–60
18. Robbins JE, Redmiles DF (2000) Cognitive support, UML adherence, and XMI interchange in Argo/UML* 1. Inf Softw Technol 42(2):79–89
19. Schenk KD, Vitalari NP, Davis KS (1998) Differences between novice and expert systems analysts: what do we know and what do we do? J Manage Inf Syst 15(1):9–50
20. Shanks G, Tansley E, Weber R (2003) Using ontology to validate conceptual models. Commun ACM 46(10):85–89
21. Siau K, Cao Q (2001) Unified modelling language: a complexity analysis. J Database Manage 12(1):26–34
22. Siau K, Erickson J, Lee LY (2005) Theoretical vs. practical complexity: the case of UML. J Database Manage 16(3):40–57
23. Tigris.org (2009) Welcome to ArgoUML. http://argouml.tigris.org/. Retrieved 21 April 2010
24. de la Vara JL, Sànchez J (2008) Improving requirements analysis through business process modelling: a participative approach. In: Business information systems: 11th international conference, BIS 2008, Innsbruck, Austria, May 2008, Proceedings, p 165
25. Wand Y, Weber R (2002) Research commentary: information systems and conceptual modelling—a research agenda. Inf Syst Res 13(4):363–376
26. Wohed P (2000) Tool support for reuse of analysis patterns—a case study. In: Conceptual modelling—ER 2000, Lecture Notes in Computer Science, vol 1920. Springer, Berlin, pp 591–660

MetaMorPic: Self-Contained Photo Archival and Presentation

Tomas Cerny and Michael J. Donahoo

Abstract Cost associated with the maintenance and scale of custom services presents one of the most significant barriers to entry for information providers. One solution proposes pushing computation into the cloud where providers like Amazon maintain a scalable, world-wide platform for virtualization. While this avoids certain hardware maintenance costs, service providers must still maintain clusters of virtual machines. To address this, contextual services like Flickr provide a complete solution; unfortunately, this limits information providers to the available services and look-and-feel. We propose a compromise solution that combines existing services and applications. Such a solution decreases development and maintenance costs by providing standardized services with third-party maintenance, while allowing customizable functionality and look-and-feel. We present a specific example of this type of blended solution in our MetaMorPic system, which provides photo archiving and presentation capabilities using third-party software and services.

1 Introduction

Public information systems are part of the critical infrastructure for distributing information. Much of this information is distributed as text, which fits well with Semantic Web [1]. The idea of Semantic Web is to provide standardized rules to structured information so that it is easy to find and process by computers [1]. More and more we also receive information in nontextual forms, including voice messages, images, videos, etc. Clearly, such data are also semantic instruments [2, 3]. Existing technology allows us to associate metadata with images, etc. that extends captured information. Many systems allow the gathering of metadata associated with images [4–8]. Such metadata should be application independent, which allows us to use images for archival purposes because we can access the metadata many years later without a need of a specific presenting application. The image format JPEG,

T. Cerny (✉)
Department of Computer Science and Engineering, Czech Technical University,
Charles Square 13, 121 35, Prague 2, Czech Republic
e-mail: tomas.cerny@fel.cvut.cz

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_12,

for example, allows us to keep all the metadata within the image [9], but its integration within systems is partial [6–8] requiring additional information to be held apart from the image. In addition to its descriptive purpose, multimedia metadata provides searchable, machine-readable access to images similar to that of Semantic Web.

In this paper, we propose a self-contained photo archival and presentation system capable of metadata gathering. Such a system is capable of capturing events, their attendees, and extensive event descriptions in textual form. We store metadata within the image files to avoid application dependence. The images can be seen after many years using different presenting tools, and the semantics can be fully reproduced from just an image and its metadata. We evaluate existing, commonly-used services and applications and provide implementations compatible with existing thirdparty vendors. We develop the application by combining existing services [8,10] and presentation tools [11–13] through a light-weight data-interchange format [14]. The application was evaluated in production for a large event, the 2010 ICPC World Finals in China, with nearly 1,000 local event participants and many more remote users, generating significant user load. The evaluation event is popular world wide, accessed by users and media [15] form various countries.

The paper is organized as follows. In Sect. 2, we introduce existing solutions and their drawbacks. The proposal, strategy for the development, and results are described in Sect. 3. An example of the use is described in Sect. 4, followed by conclusions in Sect. 5.

2 Existing Solutions and Drawbacks

Information systems are very popular these days, and multiple vendors provide free, open-source applications and web services. For our work, we look specifically at media presentation and archiving. Existing services provide a varying set of both features and interfaces. We base our media presentations evaluation on its capability to categorize media for selection and how it allows display of metadata to the user. For archiving, we look at storing the media with its metadata, how we supply the metadata and what are the requirements for the installation, maintenance and backup.

We evaluated many static media galleries [11–13] all of which provide attractive and rich user interfaces that can be customized. Various vendors [10, 7, 8, 16] provide the capability of sharing media where users can upload and organize a gallery, but these systems provide a vendor-specific appearance. We also consider Semantic Web approaches for our goals. In addition to existing system, there is the option of developing our own system. In this section, we investigate the pros and cons of various options for media presentation and archiving.

2.1 Traditional Development Approaches

The traditional approach of custom application development forces us to manage data persistence ourselves, typically in a relational database. While the developer has the freedom to specify any data schema, systems dealing with photo archiving typically produce similar schemas. Custom persistence solutions also include the onerous task of conducting, storing, and managing backups. Once persistence and backup is in place, the developer must start to design model-view-controller (MVC) [17] type application. Modern enterprise frameworks rely on object oriented design (OOD) [18] and mapping the database schema to the object domain model using some object-relational mapping framework (ORM) [19]. This mapping adds repetitive effort to development because it parallels data schema development. Having ORM in place allows us to define various queries to obtain data we wish to show to the application user. We must deal with functions for the application manager concerned with photograph management. To provide application users with suitable presentation, we must build some display view, most probably for a web browser. Similarly we can provide a potentially faster solution using a standalone application capable of a dialog with the controller over web services.

While this approach provides design freedom, we must deal with an expensive and error-prone development process, test modules, apply backups and spend time on application maintenance. One complicated issue involves application scalability where we might be required to invest in hardware resources without knowing demand. Developing a new application is very time consuming, and we may not get the features and scalability that existing systems already provide because of time constraints and resource limitations.

2.2 Existing Semantic Approaches

In Sect. 1, we suggested a relationship with Semantic Web. The research in techniques of image content annotation has long history [20–22]. We consider PhotoStaff [4, 5], a tool for annotating images on the Semantic Web. Semantic Web uses ontology languages like Resource Description Framework (RDF) [23] and Web Ontology Language (OWL) [24], but research does not address the images with their metadata. PhotoStaff uses metadata embedded in images, extract them and encodes them in RDF/XML. The tool then takes advantage of of the built infrastructure to allow search and navigate.

Although this tool is capable of advanced searches and advantages of Semantic Web, we also find several drawbacks. First, if we rename or shuffle images, we lose the metadata information, although we can partially recover from the image embedded metadata. Second, this approach binds our systems to a specific tool. Third, we do not receive any annotation automation. Finally, this system lacks the features and stability to be used in production [8, 10]. To address these problems, we propose embedding all the metadata into an image, which helps to avoid information loss. From a functional decomposition perspective, we get advantages if the system or data are broken down into units of behavior or information [25–27]. A solution based on this approach would naturally store image metadata in the image itself, and if an ontology is built, it should be build on the top of the low-level metadata.

2.3 Media Sharing Systems

Media sharing is rapidly growing in popularity on social networks or web galleries to provide access to a variety of consumers. The last few years have witnessed the rapid expansion of the social networking site Facebook [7], and we have seen cutting edge features brought by Google's Picasa [6] and its competitor Yahoo's Flickr [8]. These media sharing applications provide options to manage and organize media and metadata annotation automation. They allow image resizing, cropping, and other modification. These systems utilize media metadata [9, 28, 29] for media specific information (GPS, camera parameters and settings) and for storing a user defined information (keywords). The latest features include face recognition and detection where the application detects similar faces and we tag the set of faces by the person name. This allows us to see all media related to a particular person and detect him/her in other media. These systems provide cutting edge features, and we expect this trend to continue. The advantage of their use is that we receive all the nice features already developed (patented) and the benefit of storage, but we need not be concerned with the database definition, maintenance, and backups. Drawbacks include restricted capability of structuring the gallery, the vendor specific appearance and its binding to the vendor. Existing media sharing systems hide their internal implementation from the developer. The critical services for managing metadata exist as well as the sufficient search capabilities. The key thing for a developer is the production level and scalability.

2.4 Existing Media Gallery Presenters

Plenty of presentation galleries [11–13] can be found online, where designers provide a sleek user interface attractive to users. These galleries are often available for free and provide just the presentation without connection to any archive. Static media galleries provide just a part of the functionality we need. These are also easy to customize or modify to fit our company expectation providing specific information and references. Many modern elements that make the presentation attractive for the user utilize JavaScript (JS); consequently, most of the galleries utilize JS in either its pure form or using a JS framework [30–32]. JS not only enable a desktop-like interface but also decreases the server load by implementing a part of the presentation logic at the client side. From our perspective, we cannot solely use a static gallery, but we may consider its use in a connection with some photo archive.

3 MetaMorPic Proposal and Implementation

Previously, we introduced existing approaches and applications, summarizing their properties. Here we propose a solution to build a self-contained photo archival and presentation system, MetaMorPic.

The system should archive photographs with metadata describing the context such as who is on the picture, where was it taken, who is the author, what camera was used, what event it contains, or if there is a specific delegation or group captured by the picture. JPEG EXIF [9] and IPTC [28] formats allow us to store metadata directly in the image file. The presentation should have access to entire photo archive as well as search capabilities to select specific categories of people or events. The system should not replicate any of the information for the archival and presenter.

Our approach utilizes the services of existing archival and image presentation services by simply providing the necessary glue code to compose these services. The combination of the existing systems presents many benefits. The main issue we face is that existing applications work on their own; we need to find a connection mechanism to connect them. Some of the systems we introduced in relation to archival capabilities [7, 8, 10] provide an API that allows other programs to integrate their functionality, although often only a functionality subset of the application is available. We use these systems for photo archival and management, and via their API provide the images and metadata to the presentation subsystem. Archival services store and support basic query operations on all galleries. The galleries should be naturally structured to smaller sets of photos to perform the queries on. Presentation is covered by static media galleries [11–13], but we need to modify their static behavior.

By using the existing archival and presentations systems, we reduced the development efforts to the connection layer. The connection layer may either be implemented as a server-side application (i.e., using Java, PHP, .NET, etc.) or on the client-side (i.e., using JavaScript, Flash, etc.) [33]. The first option provides simplification for the development but adds the requirement to host the connection layer application at some server, requiring both maintenance and hardware costs. The client-side connection layer might be more challenging from the development perspective but simplifies the hosting requirements to a static web location with no need of a server-side interpreter engine. The second solution has advantages for the practical use.

Among the presentation applications, we choose Galleriffic [11], which is a jQuery plug-in for rendering rich, fast-performing photo galleries. It provides sufficient documentation for its modification and allows us to customize the appearance. Next we need to connect the client-side presentation to the archive. Specifically, we want a lightweight, data-interchange format that is supported by both client and server side so we adopt JavaScript Object Notation (JSON) [14]. This notation is well supported by the archive API's and jQuery.

For the user domain, there is a need of media categorization. Such a generic categorization is not provided by any of the archival. The solution is to define a metadata name-space for the structure we expect and use archival search capability to search over such an name-space. The archival embeds the metadata in each media

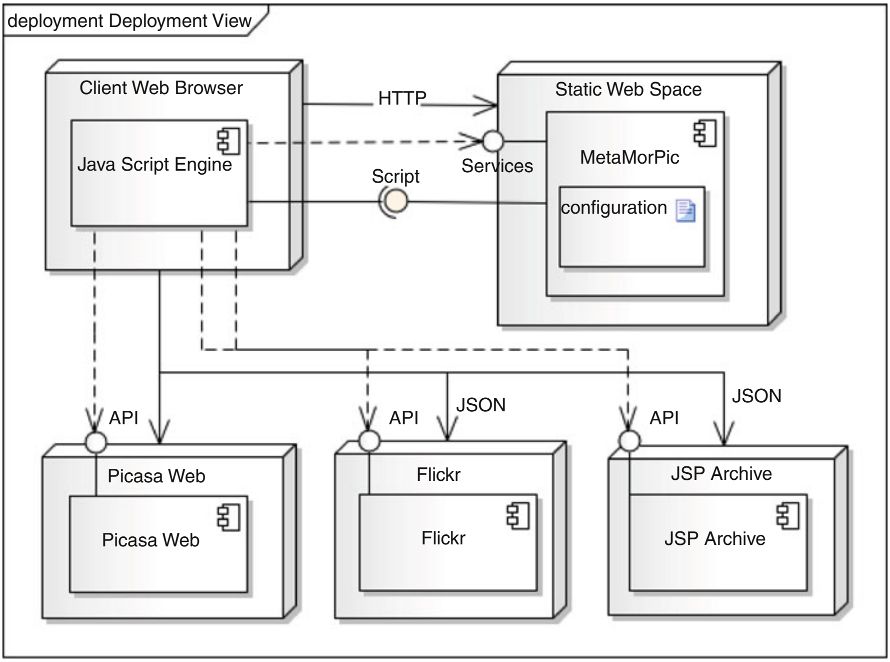


Fig. 1 Deployment of MetaMorPic

[9, 28, 29], which has the advantage that exports of the media contains the metadata. If we export all our galleries to another system, the metadata will be preserved. Currently, most systems do not include facial recognition data in the image metadata. In addition, most image archival services do not share this information. But since we can read this position out of the original archival application, we provide a Perl script[1] for porting external information to image metadata. This allows us to preserve all metadata and provide it also to the presentation subsystem. Second we encode the facial information in such a way that indexing system cannot detect this information by indexing our application to provide privacy. Besides the Picasa Web and Flickr compatibility, we also built a JSP-based archive capable of media storage and JSON services. Since our media contains all metadata, we can switch the archives with no compatibility issues. The application structure is shown by Fig. 1.

MetaMorPic [34] is an application built on the top of existing infrastructure provided by third-parties. We reduce the development effort to the connection layer to make the presentation provided by Galleriffic compatible with media archives Picasa Web, Flickr, or our own custom JSP Archive. The MetaMorPic web application contains all of its logic at the client-side, which speeds up the application reaction time. The logic is built to provide the user with pay-as-you-go experience so only

[1] In addition we work on a Java based Picasa desktop plugin which would allow to simplify the metadata port.

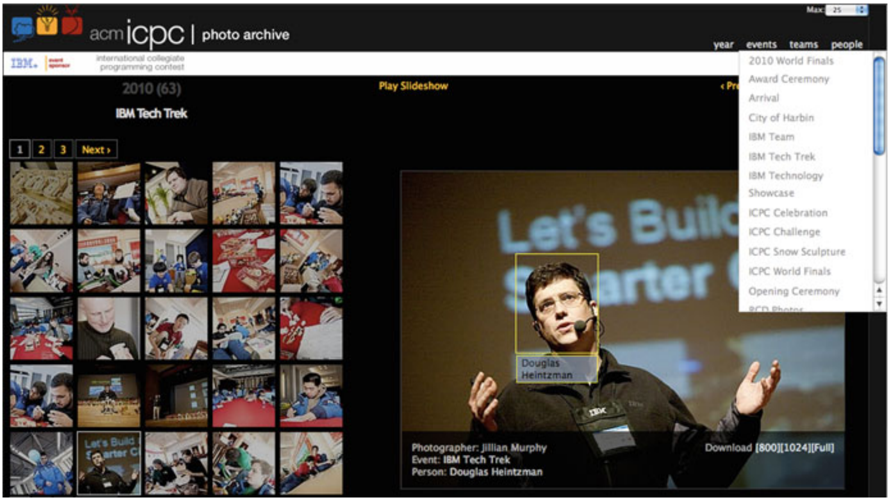


Fig. 2 MetaMorPic in producation

a small media page is provided by prefetching and the rest of the media pages are loaded on demand. Every modern web browser with enabled JavaScript can display our application. The scalability of the application is moved to the archival provider (Picasa Web, Flickr), which is generally better than what individuals can provide. This pushes the maintenance and backup responsibility to the archival provider. Using MetaMorPic does not require installation of any specialized software; instead, it simply needs a static web location for the MetaMorPic client-side code and a Picasa Web/Flickr account. Managers of Picasa Web may also gain from the use of standalone Picasa photo editing software [6] that allows the user to manage and organize the content and synchronize it with the web archive. MetaMorPic is shown in Fig. 2.

4 Example Use

To evaluate MetaMorPic, we deployed it at the 2010 ACM-ICPC World Finals in Harbin, China. This competition hosted nearly 1,000 participants with five photographers and numerous events [15, 34]. For the past few years, the media team established a process for the competition media archiving and presentation. A proprietary system was used, and the management work was tedious, mostly because every face had to be identified manually, images resized and the system did not support all management features.

Every photograph goes normally through correction and metadata tagging. We keep camera information (flash, lens, camera type, GPS), author and classification of the image. For this event, we archive who is in the image (along with face position), what event and, if applicable, which university team. With MetaMorPic, we automate the process and greatly simplify the work of the media team. The benefit is that the media team can use the Picasa's desktop photo editing software. A manager can simply tag one face, and the Picasa desktop auto-recognizes the person in the galleries.

A standard web server in Texas at Baylor University currently serves the static web page implementing the presentation aspect of MetaMorPic for ACM-ICPC. Google's cloud provides the web services and hosts the images. We have evaluated the application responsiveness and interaction on multiple web browsers in the USA, Canada, China, and the Czech Republic.

5 Conclusion

To build a production-level information system based on images and enhanced with metadata is a challenging task. Recent research in the area is moving toward the use of Semantic Web, but there are difficulties with respect to development and archival. Our proposal provides a modern approach to application development, which leverages recent advances in services while minimizing complexity and cost. Rather than building a brand new application, we propose using services provided by a third party and focus only on specifics we expect from the application. We have proposed and implemented a self-contained photo archival and presentation application called MetaMorPic. This application combines multiple services to provide many advantages of the final system and a great user experience in form of client-side web application. Our application is build on the top of the jQuery plug-in Gallerific for presentation, readily available services for archiving and searching such as Picasa Web and Flickr, and a metadata managing desktop application (i.e., Picasa). Our application does not require any server-side interpreting, just static web space. Finally, we evaluated the application in production with positive feedback from its users and managers.

Potential future work includes building a MetaMorPic plugin to Picasa desktop to simplify metadata propagation. For advertisement of the galleries, we plan to provide a virtual tour for selected photos from the competition. As there exists multiple conferences/competitions for our target organization, we plan to extend MetaMorPic functionality so it would dynamically load a gallery based on the URL GET parameter specifying the gallery name. This allows keeping a central access point and naturally scaling in terms of the gallery management. In case of the ACM-ICPC this allows to put the responsibility of archiving on the regional contest managers and keep the presentation accessible from the main web page. Finally, we plan to mesh it with other network services such as Facebook, blogs, Google maps, etc.

Acknowledgements We would like to thank Aaron Kelley for the JSP Archive development and Hans Domjan, the ICPC Photo Management Director, for his feedback and suggestions. Our thanks goes to the ACM-ICPC for supporting this project.

References

1. Cross P, Miller L, Palmer S (2001) Using RDF to annotate the (semantic) web. http://ilrt.org/discovery/2001/07/KCAP-ann
2. Confucius http://en.wikipedia.org/wiki/A_picturejs_worth_a_thousand_words
3. Zhao J, Klyne G, Shotton D (2008) Building a semantic web image repository for biological research images, Department of Zoology, University of Oxford, 5th European semantic web conference
4. Halaschek-Wiener C, Golbeck J, Schain A, Grove M, Parsia B, Hendler JA (2005) PhotoStuff an image annotation tool for the semantic web. In: Gil Y, Motta E, Benjamins VR, Musen MA (ed) Posters 4th international semantic web conference, Galway, Ireland, Nov
5. Halaschek-Wiener C, Golbeck J, Schain A, Grove M, Parsia B, Hendler J (2006) Annotation and provenance tracking in semantic web photo libraries, University of Maryland, Springer, Berlin, Heidelberg
6. Google Picasa—photo management application (2010) http://picasa.google.com
7. Facebook—social network (2010) http://facebook.com
8. Flickr—online photo management and sharing application (2010) http://flickr.com
9. Exchangeable Image File Format (EXIF) (2010) http://www.exif.org
10. Google Picasa Web Album—photo sharing application (2010) http://picasaweb.google.com
11. Galleriffic—jQuery plugin (2010) http://www.twospy.com/galleriffic
12. Galleria—JavaScript image gallery (2010) monc.se/galleria/demo/demo_01.htm
13. Imago—JavaScript image gallery (2010) http://imago.codeboje.de
14. JavaScript Object Notation (JSON)—lightweight data-interchange format (2010) http://www.json.org
15. NYTimes—article reference to example event, source MetaMorPic (2010) http://www.nytimes.com/2010/02/22/technology/22cyber.html?scp=1&sq=jillian%20murphy
16. Kodak—photo sharing application (2010) www.kodakgallery.com
17. Erich G, Helm R, Johnson R, Vlissides JM (1995) Design patterns: elements of reusable object-oriented software. Addison-Wesley, New York, NY, pp. 395. ISBN 0201633612
18. Larman C (2004) Applying UML and patterns: an introduction to object-oriented analysis and design and iterative development (3rd ed). Prentice Hall PTR, New Jersey, NJ. ISBN:0131489062
19. Java Persistence API (2010) http://java.sun.com/developer/technicalArticles/J2EE/jpa
20. Dupplaw D, Dasmahapatra S, Hu B, Lewis P, Shadbolt N. Multimedia (2004), Distributed Knowledge Management in MIAKT. ISWC 2004 Workshop on Knowledge Markup and Semantic Annotation, Hiroshima, Japan, Nov
21. Addis M, Boniface M, Goodall S, Grimwood P, Kim S, Lewis P, Martinez K, Stevenson (2003) A SCULPTEUR: towards a new paradigm for multimedia museum information handling. 2nd international semantic web conference, pp 582–596
22. Hollink L, Schreiber G, Wielemaker J, Wielinga B (2003) Semantic annotation of image collections. Knowledge capture—knowledge markup & semantic annotation workshop
23. Resource Description Framework (RDF) (2010) http://www.w3.org/RDF
24. OWL Web Ontology Language (2010) http://www.w3.org/TR/owl-features
25. Parnas DL (1978) Designing software for extension and contraction. In: Proceedings of the 3rd international conference on software engineering, pp 264–277
26. Parnas DL (1974) On a 'Buzzword': hierarchical structure. In: Proceedings of the IFIP congress 74, pp 336–339
27. Parnas DL (1972) On the criteria to be used in decomposing systems into modules. Commun ACM 15(2):1053–1058
28. IPTC Photo Metadata (2010) http://www.iptc.org/cms/site/index.html?channel=CH0089
29. Extensible Metadata Platform (XMP) (2010) http://www.adobe.com/products/xmp
30. Scriptaculous—JavaScript library (2010) http://script.aculo.us/

31. jQuery—JavaScript library (2010) http://jquery.com
32. Prototype—JavaScript framework (2010) http://www.prototypejs.org/
33. Fowler M (2002), Patterns of enterprise application architecture. Addison-Wesley Longman Publishing Co., Inc., New York, NY, ISBN:0321127420
34. MetaMorPic: Self-contained photo archival and presentation (2010) http://cm.baylor.edu/gallery/MetaMorPic.html

A Real Time Multiagent Information Systems Architecture for Actively Monitoring Chronic Diseases

Emma Chavez-Mora and Gavin Finnie

Abstract In e-health, data warehouse systems have become an important strategy to integrate heterogeneous data sources. Data warehouse technologies are required for telecare, drug interaction for patients with multiple symptoms, electronic prescriptions and emergency datasets, in which a massive monitoring of the processes in real-time enables proactive response to exceptional and time-sensitive critical situations. However, current data warehousing and business intelligence technologies do not directly address time sensitive monitoring, real time data processing and the adaptation of data to analytical requirements. Typically, a data warehouse (DW) is designed and built on the basis of historical information. In this paper a new information system design is described for a real time adaptive data warehouse architecture that covers the process of predicting and responding to e-health requirements in order to decrease the time of reaction and enable active responses from e-health actors and operative systems.

1 Introduction

Chronic diseases are diseases of long duration and generally slow progression. These include heart disease, stroke, cancer, chronic respiratory diseases and diabetes. The World Health Organization [19] states that "chronic diseases are by far the leading cause of mortality in the world, representing 60% of all deaths". The number of people, families and communities that are affected is increasing and the problem needs to be addressed urgently.

Chronic disease management is time sensitive. Technology enables doctors, patients and the entire health care staff to respond to a crisis and prevent problems by providing early indications of deterioration in the patient.

E. Chavez-Mora (✉)
Bond University, Robina, QLD, Australia
e-mail: emchavez@bond.edu.au

Universidad Catolica de la Ssma. Concepcion, Concepcion, Chile

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_13,

In health care, various components are connected and sometimes overlapped [7]. Clinical data is stored in heterogeneous and distributed repositories across regional and international institutions around the globe [15]. Thus, moving into a specific health care domain, clinical data warehouses (CDW) can facilitate access to data and the multidimensional analysis of the data obtained in the patient care process which improves the quality of decision making. In addition, a CDW helps with the consolidation of data from multiple sources of information which assists with the extraction of only that data which is valuable for patient monitoring [15].

In a scenario in which electronic patient data travels from different organizations such as public hospitals, general practitioners, private clinicians, government and insurance payers, a robust architecture that enhances data sharing without duplication is required. Thus, the integration of information from heterogeneous sources, security access to patient's data (security infrastructure) and the access to information in a timely manner (real time) are some of the issues to solve in the clinical data warehouse domain.

Thus, we propose a solution to manage patients with heart disease by designing an architecture that: learns and reveals the disease activity patterns through day to day measurements and the clinical history of the patient, reacts in real time by sending alarms according to changes in those patterns, and is adaptive to new system conditions and changes in health care requirements.

In the following sections a description of the framework designed to manage clinical data is presented. The research being undertaken fits into the design science paradigm with the development of an artefact to validate the approach. An overview of the material and methods of the framework design is described in the following section. Then, an explanation of the base of knowledge and application architecture is given. Finally, we summarize the results and suggest directions for further research.

2 Framework Design

DW technologies are used to facilitate the analysis of data that comes from different data sources. Here data extraction, transformation and loading are essential components to build a data warehouse. Nevertheless, data extraction to provide rapid response is an open area of research. Solutions are focused on the use of techniques such as data streaming and cache memory [13].

However, all of the techniques already in use have to schedule the data extraction in some way; therefore it is the data warehouse which is querying for data. The major challenge is to provide dynamic interaction between users and service providers. But the state of the art of distributed systems administration (data bases for example) is still dominated by human administration and intervention. Therefore, it may be possible to enable local autonomy at the level of the data sources to push information (transfer data) to the warehouse. Thus, as soon as "valuable" data arrives at the data source, it will immediately be sent to the warehouse. To enable data push, agents will be used as these are defined as entities that enable local intelligence to react in particular environments.

The research in this paper focuses on solving the information systems challenge of real-time information extraction from different sources, such as sensors and various databases, to build a real time data warehouse. Through this way, the data warehouse architecture proposed will be constructed based on the view that the decision making process needs to be active as it must include historical and new business information to perform correct and strategic decisions.

The framework applies to e-health and information system research foundations by designing an architecture that can actively monitor heart disease patients as well as providing data security and patient data analysis to support clinical decision making. To support the real time data transfer and analysis an event driven distributed agent platform is proposed. Thus, through a push system the sources of information (hospital data bases, GP data bases and sensor devices) are empowered by individual agents to control patient data and to deliver information to the learning repository that will administer alerts and patient data updates.

Current CDW approaches rely on centralised control of data updates with the timing and processing of data retrieval from disparate sources being directed by the CDW itself. Our approach empowers each source of information with sufficient intelligent processing capability to send relevant and time-critical data to the CDW as it becomes available.

The first step to develop the architecture of an e-health real time data warehouse is to define the flow of information from the warehouse and the analysis requirements. To support the real time data transfer and analysis an event driven distributed agent platform is required. The methodology for the architecture design was divided as follows:

- Information architecture: The data analysis and harmonization to determine which kind of information is stored in each data source was performed. An abstraction of the information requirements of the data architecture with a high-level logical representation was developed. The base of knowledge in heart disease, the relationship between the symptoms and kind of data storage in each source was then determined.
- Application architecture: The setting of rules (events) for data transfer was determined. Then, the characteristics, roles, interactions and distribution among multiple agents were described.

In this paper we discuss the data warehouse approach and only the application architecture will be discussed.

2.1 Defining a Base of Knowledge for Heart Disease

According to [2] the diagnosis of heart disease can be made on the basis of the five finger approach. This includes a careful detailed history, electrocardiogram, laboratory tests, physical examination and chest x-ray. In addition to that, the most common symptoms and features of patient with heart disease include chest pain and discomfort, breath shortness and palpitations among others. These provide around 40 factors relevant to diagnosis

Having so many factors to analyse and identifying the important features of the disease while ignoring those which confuse the diagnosis may smooth the progress of physicians to make prompt diagnosis and treatment decisions. To deal with expert knowledge considerations, diagnosis systems have been inspired by the use of intelligent techniques to detect patient symptoms (abnormal and normal functioning) at an early stage.

In the last decades several tools and data mining methodologies have been applied to develop medical decision support systems to effectively diagnose heart disease. Some of these are [3–6, 10–12, 14]. These systems commonly involve an active interaction among various medical knowledge modalities.

A different approach was followed by [20]. In this research the main clinical features to consider for heart disease diagnosis systems were found. Using a real-coded genetic algorithm 40 clinical features of heart disease were reduced to a list of 24. This new list represents the critical diagnostic features for 5 major heart diseases and should be considered to be the most relevant for diagnosis. This includes diagnostic features such as; Age, dizziness, cyanosis, chest pain, dyspnea, blood pressure and edema of lower limb.

With the features defined by [20] an overview of patient's heart disease can be described. Therefore, they have been established as the base of knowledge for our framework. An assessment of the interaction between symptoms and their level of importance for a patient with heart disease has been obtained by the use of surveys in which 5 cardiologists participated.

3 Application Architecture

3.1 Patient Data Cleansing, Transformation and Transference

In real time business process monitoring a large amount of data is frequently generated. Therefore, a new type of data management is necessary to perceive, query and retrieve continuous data streams. The automatic analysis can be achieved via mining models and the data extraction from the different data sources using streaming data management.

The data process update is generally performed according to the system administrator criteria, who defines the events and conditions in which a query or data request will be necessary. Nevertheless, to be effective, real time business monitoring needs to extract real time information. Therefore, traditional approaches are limited in their ability to provide real time information, this is why event-driven data integration, real time stream data management and real time learning capabilities for adaptation to new requirements are required [1, 8, 9, 17].

Moving into the health care arena, patients, health care providers, payers and government demand more effective and efficient health care services, and the health care industry needs innovative ways to re-invent core processes. As well as in business, health care requires real time patient monitoring, detection of adverse events, and adaptive responses to breakdown in normal processes. Early automated feedback prevents process breakdown [18].

To achieve real time data extraction, our design specifies a framework that does not rely on traditional extraction, transformation, and loading techniques (ETL) to feed the warehouse. A transference, transformation and loading approach is proposed instead (TTL). Thus, to empower and enable local autonomy at the level of the data sources to push information (transfer data) to the warehouse, agents are used as these are defined as entities that enable local intelligence to react in particular environments.

Agents are designed to perform specific goals such as monitoring a particular patient, sending alert messages to a doctor if the patient data is not within normal ranges and even evaluating the possibility of updating the knowledge because new disease information for that patient has become available. Thus, they have learning process capabilities to automatically build up learning models to be used to answer what if questions and to respond to particular patient events. Intelligent agents are used to manage business activities capturing events and monitoring the state of workflow tasks and resources.

This is a patient centred framework. Consequently the alerts and the learning process focus on those features relevant to the patient. The agents in our design have the goals of monitoring a patient's state in relation to the metrics defined as the base of knowledge (as discussed in the section of information architecture) and additionally they are responsible for checking a new entry and changes made on a patient's data and disease data. As soon as a new entry has been received from the source the agent will compare and react if the new data is valuable data for the patient and for the disease monitoring activity.

Thus, the agent alerts relevant authorities by sending the new information to the data warehouse. Through this way, a real time data stream is achieved because valuable real time data will is retrieved by the agent and sent to the data warehouse according to data extraction rules and defined events. Those rules and events come from evidence-based guidelines for heart disease and the risk scenarios of each patient.

3.2 Learning Repository-Data Warehouse

A health care enterprise includes people, systems and processes which operate under a complex adaptive environment and the field of adaptive system theory provides fertile ground for research on health care process improvement strategies [18]. Data mining of historical information can generate new insights for process intervention.

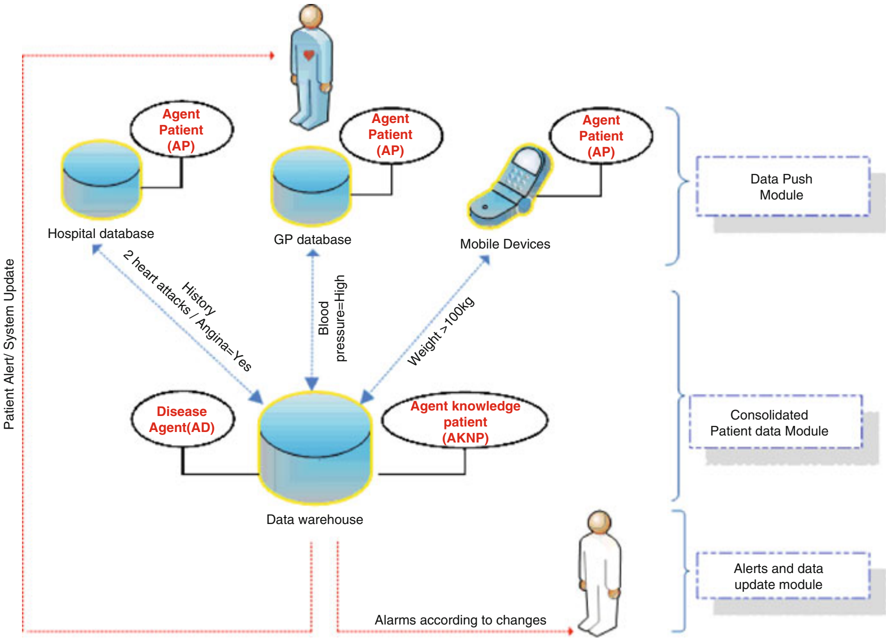


Fig. 1 Real time active monitoring architecture

As can be seen in Fig. 1, all the sources of information are actively monitored using a distributed multi agent based platform.

- *Data push module:* This platform is responsible for monitoring the patient disease data obtained in each source. Through a push system, the sources of information (hospital data bases, GP data bases and sensor devices) are empowered by individual agents (Agent patient—AP) to control patient data. AP is subscribed to the patient ID in each source of information and, through the set of specific rules that it has from the patient, it takes the decision to deliver information to the learning repository.

  To monitor patients and to update the knowledge about a patient and the disease, a case based reasoning approach was selected. Case based reasoning provides adaptation to our system because of the ability to support the automatic acquisition of subjective knowledge by adding new cases or adapting old solutions to meet new requirements. If we consider that the general knowledge of physicians is acquired from the past experience with cases they have treated themselves, colleagues, or book information, the thinking process of physicians defines the typical case.

  Thus, AP has a set of patient cases for which it is necessary to provide an alert. Each case has been built based on the historical information of the patient.

  Then when a set of data is extracted from the sources of information it will be compared with the actual cases to decide the possible outcomes. If the set of data does not match any of the cases that AP has, the information is then compared with the full base of knowledge of patients that resides in the Agent knowledge patient—AKNP. Information is only delivered when the data obtained in the source of information is not within normal ranges, does not match any of the cases obtained from the historical information, or there is not enough information to take a decision. Therefore, it represents valuable data to control the patient disease prognosis and may need to be analysed as a whole view in AKNP, Agent disease—AD and/or in the learning repository.

- *Consolidated patient module:* In this module, knowledge agents AKNP are continuously learning from the data to have the capability of adapting to new health care conditions or requirements for a particular patient. Patient prognosis decision systems present characteristics such as duality of objectives, subjective knowledge, automatic acquisition and system integration. Case based reasoning systems in medical diagnosis adapt old solutions to interpret new situations by using knowledge-based learning to reuse its own experience [4].

  As is shown in Fig. 1, there is only partial information received from each information source. There may therefore not be enough information to take action when the data received is not within normal ranges for that patient but the base of knowledge says that it is important information to consider for the disease. So, information is sent to the learning repository to take an action with a more complete view. In the learning repository AKNP will react from the stored experience to decide whether the information received from the multiple sources of information implies an alert. If the information is unknown it may be that a new case needs to be built and it will need to be contrasted with the base of knowledge of AD. An alert to the health staff will be sent anyway.

  In our design an agent has the base of knowledge to react to different events and by reasoning through experience. Then in the learning repository medical data is evaluated in AKNP and in AD to decide whether an alert should be sent or knowledge (agent rules) for a particular patient should be updated. These rules are stored to be used for the disease prognosis, to formalize the cases and to construct new cases as additional information became available [16].

- *Alerts and data update module:* Once data is received in the repository it will be analysed to make decisions such as sending alerts to health care users, or sending an alert to the agents in the sources of information because they may need to change their base of knowledge because patient metrics or conditions have changed.

4 Conclusions and Future Research

We have developed a new approach to design an active data warehouse system in which traditional extraction, transformation and loading tools do not apply. As the data warehouse is not programmed for querying the sources for the information, the transformation, filtering and information cleaning will be performed by agents.

In traditional consolidated systems, data extraction is performed for thousands of records before proceeding with data analysis. If there is not enough information to proceed with the analysis, it will only be detected once data has been collected and sent to the central repository. In our local approach, there is enough intelligence to only select relevant data for each patient to monitor and to alert if inconsistencies are found. If there is enough information, the patient agent will locally perform analysis and trigger an alert. Agents are continuously monitoring and are programmed to selectively send (by pushing) valuable information to the different architecture entities to improve and create real time decision making.

As can be seen, the design of the architecture discussed in this document focuses on solving three research areas.

- Access to information in a timely manner and real time data warehouse alerts and response: We are using an event driven approach as a way to sense and react in real time to certain environment conditions. Thus, sources of information react by pushing data into the warehouse. Real time response is kept to a minimum latency by eliminating the data availability gap which will enable organizations to concentrate on accessing and processing valuable data.
- Adaptation to new patient requirements: A data warehouse architecture capable of learning from the environment (disease) is designed as the way to automatically update the events that trigger the systems alarms (patient's medical conditions). Therefore, this design enables logical adaptability of the data warehouse content which is adaptation of the knowledge rather than data warehouse structure.
- The gap of real time intelligent architecture for active monitoring is addressed by providing intelligent capability to determine the triggers that will push the data to feed the data warehouse.

Nevertheless, although we have chosen a case based reasoning methodology for the update frequency of the learning process we have not proved the real time response of this architecture at this stage. It may consider updating as a new input is received or if a pre-specified rule is triggered. Information will then be communicated to other agents and to the data warehouse which will be able to mine complex real-world data of patients based on knowledge engineering techniques discussed above.

References

1. Azvine B, Cui Z, Nauck D, Majeed B (2006) Real time business intelligence for the adaptive enterprise. In: Proceedings of the 8th IEEE international conference on e-commerce technology and the 3rd IEEE international conference on enterprise computing, e-commerce, and e-service, pp 29–40
2. Chizner M (2002) The diagnosis of heart disease by clinical assessment alone. Dis Mon 48:7–98
3. Chohra A, Kanaoui N, Madani K (2007) Hybrid intelligent diagnosis systems. In: Proceedings of the computer information systems and industrial management applications, pp 115–120
4. Chowdhury S, Birswas A, Chowdhury R (2008) Design, simulation and testing of an optimized fuzzy neuronal network for early critically diagnosis. In: IEEE first international conference on emerging trends in engineering and technology, pp 665–670

5. Comak E, Arslan A, Turkoglu I (2007) A decision support system based on support vector machines for diagnosis of the heart valve diseases. Comput Biol Med 37:21–27
6. Das R, Turkoglu I, Sengur A (2009) Effective diagnosis of heart disease through neuronal network ensembles. Exp Syst Appl 36:7675–7680
7. Ewen E, Medsker C, Dusterhoft L (1999) Data warehousing in an integrated health system: building the business case. In: 1st ACM international workshop on data warehousing and OLAP, pp 47–53
8. Ferguson M (2005) Building intelligent agents using business activity monitoring. Inf Manag Mag
9. Greiner T, Düster W, Pouatcha F, Von Ammon R, Brandl H, Guschakowski D (2006) Business activity monitoring of Norisbank taking the example of the application easyCredit and the future adoption of complex event processing (CEP). In: Proceedings of the 4th international symposium on principles and practice of programming in Java, pp 237–242
10. Huang M, Chen M, Lee S (2007) Integrating data mining with case-based reasoning for chronic diseases prognosis and diagnosis. Exp Syst Appl 32:856–867
11. Jianxin C, Xing Y, Wang J, Zheng C (2007) An unsupervised multi-valued stochastic neural network algorithm to cluster in Coronary Heart Disease data. In: Proceedings of the 2007 international conference on convergence information technology, pp 640–644
12. Kim K, Choi H (2007) Design of a clinical knowledge base for heart disease detection. In: Proceedings of the 7th IEEE international conference on computer and information technology, pp 610–615
13. Nguyen T, Schiefer J, Tjoa A (2005). Sense & response service architecture (SARESA): an approach towards a real-time business intelligence solution and its use for a fraud detection application. In: Proceedings of the 8th ACM international workshop on data warehousing and OLAP, SESSION: data warehouse design 2. Bremen, Germany, p 7786
14. Ribeiro M, Traina A, Traina C, Rosa N, Marques P (2008) How to improve medical image diagnosis through association rules: the IDEA method. In: 21st IEEE international symposium on computer-based medical systems, pp 266–271
15. Sahama T, Croll P (2007) A data warehouse architecture for clinical data warehousing. In: First Australasian workshop on health knowledge management and discovery, vol 68, pp 227–232
16. Salem A, Roushdy M, Hodjod R (2003) A case based expert system for supporting diagnosis of heart disease. In: Proceedings of the international congress for global science and technology, ICGST
17. Soualmia L, Darmoni S (2005) Combining different standards and different approaches for health information retrieval in a quality-controlled gateway. Int J Med Inf 74:141–150
18. Sutherland J, Van den Heuvel W (2006) Towards an intelligent hospital environment: adaptive workflow in the OR of the future. In: Proceedings of the 39th annual hawaii international conference on system sciences vol 5, pp 100–102
19. World Health Organization (2008) Report of preventing chronic diseases: a vital investment. http://www.who.int/chp/chronic_disease_report/contents/en/index.html. Accessed 18 June 2008
20. Yang H, Zheng J, Jiang Y, Peng C, Xiao S (2008) Selecting critical clinical features for heart diseases diagnosis with real-coded genetic algorithm. Appl Soft Comput 8:1105–1111

Making Business Systems in the Telecommunication Industry More Customer-Oriented

An Analysis of Real-Life Transformation Projects

Christian Czarnecki, Axel Winkelmann and Myra Spiliopoulou

Abstract Market changes have forced telecommunication companies to transform their business. Increased competition, short innovation cycles, changed usage patterns, increased customer expectations and cost reduction are the main drivers. Our objective is to analyze to what extend transformation projects have improved the orientation towards the end-customers. Therefore, we selected 38 real-life case studies that are dealing with customer orientation. Our analysis is based on a telecommunication-specific framework that aligns strategy, business processes and information systems. The result of our analysis shows the following: transformation projects that aim to improve the customer orientation are combined with clear goals on costs and revenue of the enterprise. These projects are usually directly linked to the customer touch points, but also to the development and provisioning of products. Furthermore, the analysis shows that customer orientation is not the sole trigger for transformation. There is no one-fits-all solution; rather, improved customer orientation needs aligned changes of business processes as well as information systems related to different parts of the company.

1 Introduction

Telecommunication markets and companies have been subject to tremendous changes over the last decades (Ahn and Skudlark 2002). Companies are confronted with increased competition, accompanied by price reductions and short innovation cycles (Mikkonen et al. 2008). From a marketing perspective, innovative products have to be flexible and swiftly taken to the market (Knightson et al. 2005). From the production perspective, cost reductions have to be guaranteed (Bruce et al. 2008). Both are combined with the shift of usage patterns from classic fixed-network telephony to mobile services (Peppard and Rylander 2006) and verticalization strategies (Fink and Markovich 2008). Due to the increased competition, customers are more sensitive to service quality, when compared to former times where services of big monopolists

C. Czarnecki (✉)
Detecon International GmbH, 53227, Bonn, Germany
e-mail: christian.czarnecki@detecon.com

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_14,

were the only way to get a phone connection (Knightson et al. 2005). Especially these former monopolists are struggling to change their historically grown structures towards the customer. We have observed that all these challenges have lead to many different transformation projects with the objective to change the business processes, information systems, or both. The starting point and topical focus of these projects vary, e.g. network oriented roll-out of IP-based technologies (Choi and Won-Ki-Hong 2007), streamlining production structure and systems (Bruce et al. 2008) as well as new value chains to realize innovative product bundles (Mikkonen et al. 2008).

Beside the different foci of such projects, the alignment of company strategy and processes with the technology and the changes technology implies is a crucial success factor for new Enterprise Architecture implementations (Österle and Blessing 2003). So far, research has mostly focused on network-related aspects (e.g., Knightson et al. 2005; Lee and Knight 2005; Grida et al. 2006) as well as on specific questions related to information systems (Bertin and Crespi 2008) and productions structures (Bruce et al. 2008). However, there is little work about methodical approaches towards addressing the transformation needs of telecommunication companies and covering all of information systems, business processes and strategy with a clear setting of priorities.

The literature provides many indicators that customer orientation is a good motivation for transformation, especially in the context of customer loyalty. In the marketing research the relationship between customer satisfaction and loyalty has been intensively discussed (Fornell et al. 1996; Mittal and Kamakura 2001). Customer satisfaction is defined as the individual evaluation of the performance on an offering at a specific date (Johnson and Fornell 1991). It is built-up over time and mediates the effects of product quality and price on loyalty (Bolton and Lemon 1999; Fornell et al. 1996). In this context loyalty is explained by affective commitment (e.g. personal interaction, trust) and calculative commitment (e.g. switching costs) (Bendapudi and Berry 1997).

Existing research in the telecommunication industry discusses customer orientation based on qualitative analysis of customer surveys or by quantitative analysis of churn (Gustaffson et al. 2005). But from a transformation perspective, different parts of an enterprise are related to the orientation towards the end-customer. For example, the quality-of-service has a direct impact on the customer *during* the usage of a communication service. It can be improved by a change of the physical network from classical "wired" technologies to more flexible IP-based technology that allows for additional services. But also the sales process itself influences the perception of the customer. An efficient sales process does not necessarily require changes in the back office technology but needs adaption of the front-end information systems. Therefore, we started an exploratory analysis to address the following question:

How are processes and information systems affected by a transformation towards stronger customer orientation?

The data sources for this analysis are real-life cases that were collected and published by the TM Forum. This is an international non-profit organization that comprises telecommunication companies, equipment and software suppliers, solution integrators, consulting companies and research organizations. It provides strategic guidance and practical solutions to improve the management and operation of telecommunication companies.

We selected 38 cases that are dealing with customer orientation by performing a full text search. The case analysis follows mainly the methodology proposed by Schubert and Williams (2009). We performed an open coding based on a holistic structure of transformation projects in the telecommunication industry. Research based on case studies can be applied to phenomena which are dynamic in nature and have not yet been fully developed and established (Yin 2003). So this method is appropriate for obtaining insights about the improvement of customer orientation.

The rest of the paper is organized as follows. In Sect. 2, we discuss related work on transformation of telecommunication companies, starting with the approach of "Enterprise Architecture" and continuing with a telecommunication specific approach. Our work builds upon the holistic framework proposed by Czarnecki et al. (2009), which we describe briefly in the same section. In Sect. 3, we describe our approach for the case study analysis. In Sect. 4, we discuss our results and derive findings for our research. The last Section summarizes our study, discusses limitations and gives an outlook on further research steps.

2 Framing the Transformation of Telecommunication Companies

The need for coordinating strategic positioning, organizational structures and business processes on the one hand and IS design on the other hand have been intensively studied in the literature, see e.g. Frank (2002), Winter and Fischer (2007), Zachmann (1997). The concept of "Enterprise Architecture" (EA) has been proposed to support this coordination task on various levels of abstraction (e.g. Urbaczewski and Mrdalj 2006; Winter and Fischer 2007; Zachmann 1997). On the basis of the ANSI/IEEE Standard 1471-2000, EA can be considered as a fundamental structure of an organization, its individual elements and their relationships to one another and to the environment (Winter and Fischer 2007). Most frameworks distinguish among the following layers (Winter and Fischer 2007): business architecture, process architecture, integration architecture, software architecture, technology (or infrastructure) architecture.

Beside these generic approaches of EA, a specific approach for telecommunication companies was proposed by the TM Forum (Reilly and Creaner 2005): the "Solution Framework". Its objective is to enable telecommunication companies to analyze their business processes and systems against industry standards as well as to support the development and implementation. It provides a methodology and a repository of documentations, models and guidelines. The "Solution Framework" consists of the following, according to Reilly and Creaner (2005):

- *Business Process Framework:* it provides a map and common language of business processes that are used in the industry.
- *Application Framework:* it considers the roles and the functionalities of the various applications.

- *Information Framework:* it provides the concepts and principles needed to define a shared information model.
- *Systems Integration Framework:* it defines the architectural principles needed to create components that operate successfully in a distributed environment.

The "Solution Framework" is mainly oriented towards a definition of a common language between telecommunication companies, software vendors and system integrators. From the EA perspective (e.g. business architecture), some topics are missing. Nonetheless, compliance with the "Solution Framework" is essential to ensure compatibility to standard IT-products as well as interoperability of distributed solutions by different vendors. But the "Solution Framework" alone is not sufficient to understand a transformation project. Therefore, we are following both the enterprise architecture methodology as well as the "Solution Framework".

We consider innovative technologies in the telecommunication industry as transformation enablers and even promoters. We therefore use as basis for our approach the framework proposed by Czarnecki et al. (2009), where the Next Generation Network (NGN) technology is observed as a trigger for transformation in the telecommunication industry: the authors frame the transformation in the context of strategy, business processes and information systems. We believe that this framework is applicable beyond the example of NGN, so we use their framework with minor adjustments of the layer "business processes". We have added processes for strategic and enterprise management tasks to cover the whole company.

It follows a concise overview of the framework (cf. Fig. 1) as basis for our analysis: telecommunication companies can be roughly structured into the three layers *strategy*, *business processes* and *information systems*. Appropriate objectives from the marketing perspective for the *strategy* layer are reduced churn, reduced time-to-market and increased flexibility of customer products. They reflect the marketing

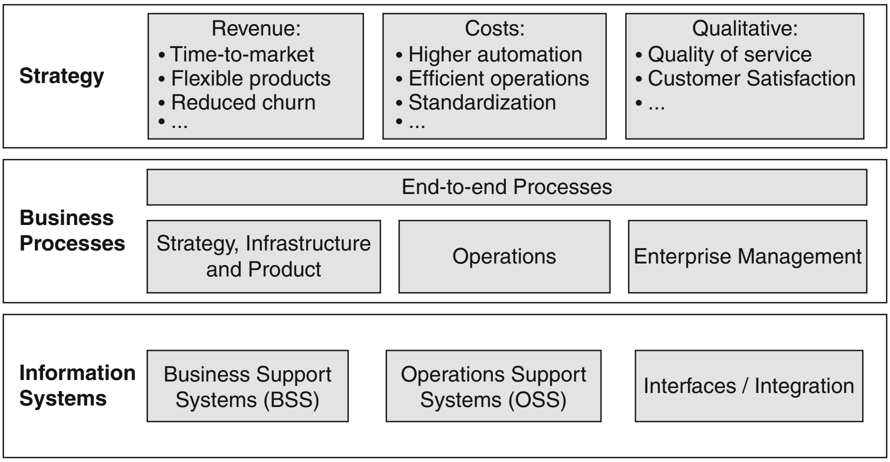


Fig. 1 Framework to structure the transformation of telecommunication companies (Czarnecki et al. 2009)

perspective, thus emphasizing the central role of customers as a driving force of the transformation process. We link these objectives to the superordinate strategic goal of increase in revenue. Taking the viewpoint of production, the objectives of increased automation and optimized processes are linked to the reduction of costs. Beside these quantitative objectives also qualitative objectives like increase in quality of service or customer orientation are important.

Business processes of telecommunication companies can be structured into three categories with different topical foci and different stakeholders (Reilly and Creaner 2005):

- *Strategy, infrastructure and product* processes encompass mainly internal tasks related to the offering of products. These processes are mainly executed by employees with inputs from the market and supplier/partner.
- *Operations* processes are covering all tasks for sales and delivery, usage and after-sales. These processes are mostly triggered by the customer.
- *Enterprise management* processes are a collection of all necessary support processes (e.g., accounting, human resources). These processes are focused on all stakeholders except customers.

Beside this, an end-to-end process view is used to link the different process areas with each other.

From the *information system* perspective telecommunication companies can roughly be structured into the business part (pre-sales, sales and after-sales) and the operational or production part (development and provisioning of products). Therefore, their IT-systems are normally divided into business support systems (BSS) and operational support systems (OSS). In the past, most telecommunication companies have developed and introduced new products, as result of technological innovations. For new technologies, new OSS were created. To integrate these new systems in the overall IT architecture, interfaces between different OSS and BSS were developed. This leads to complex and silo-oriented IT-structures. Today, telecommunication companies try to overcome these silo-oriented structures for instance with service-oriented architectures (SOA). As automation and cost reduction are major objectives, the use of standard IT products—so called commercial-of-the-shelf systems (COTS)—is a mandatory requirement for most telecommunication companies. Therefore, a challenge is the mapping of the process requirements to standardized IT-services delivered by COTS. The analysis of case studies in the next section contributes to identifying the requirements posed on IT-services from the specific business perspective of customer orientation.

3 Analysis of Cases Studies

Our objective is the analysis of projects in the telecommunication industry to understand which parts are related to a transformation towards an improved customer orientation. The following section describes the design and methodology of the analysis and the TM Forum cases used in it.

3.1 Design and Methodology of the Analysis

To describe a transformation, all changes from the *as-is situation* to the *target situation* have to be described (Österle and Blessing 2003). Therefore, we base our analysis on the existing approaches of Enterprise Architecture (cf. Sect. 2) as a holistic methodology to describe the transformation of an enterprise. In particular, we analyse two key areas. First, we have to understand why telecommunication companies aim at changing the as-is situation, the so-called "strategic driver". To understand the transformation itself we need to identify those parts of the company that are changed. Hence, we aim at analyzing the strategic drivers of these projects, which results into the following question:

Q1: Is customer orientation the sole driver of a transformation?

The second question addresses the area of change. Hence, an analysis of the specific aspects that are related to each EA layer and each area is necessary:

Q2: Which parts and aspects of an enterprise are changed to achieve customer orientation?

The research follows the methodology proposed by Schubert and Williams (2009):

1. *Initialization phase:* review of literature to motivate the research and formulate the research question
2. *Exploration phase:* selection of cases and coding based on text analysis; coding is conducted by different persons independently and conflicts are discussed
3. *Consolidation phase:* consolidation of results

With help of the literature review (cf. Sect. 2), we have first categorized the cases across three dimensions: "area", "EA layer" and "aspect". The "area" refers to the issue addressed in the case study: (a) an encountered *problem*, (b) a proposed *solution*, (c) an identified *benefit*. The "EA layer" refers to the affected company layer that is mentioned in the case study. In accordance to the high-level framework (cf. Fig. 1), we consider the layers *strategy, business processes* and *information systems.* Hence, an encountered *problem* may affect the company's *business processes*; an identified *solution* at the *information systems* layer may result in *benefits* for *business processes* and/or *strategy.* The dimension "aspect" refers to the particulars within the "EA layer" and is specific to each layer. The coding framework for our analysis, including example aspects, is depicted in Fig. 2.

Area
Problem
Solution
Benefit

EA Layer	Aspect
Strategy	costs, revenue, quality,...
Business Processes	end-to-end, SIP, Operations, performance mgmt.,...
Information Systems	OSS, BSS, network, SOA, performance mgmt. system,...

Fig. 2 Coding framework for the analysis of the case studies

In accordance to Schubert and Williams (2009), we used an iterative process to define the coding categories: a first set of codes was based on the literature review and revised after a preparatory analysis. To assure qualitative and unbiased results, a parallel coding and analysis of different persons involved in the research were used (Schubert and Williams 2009).

3.2 TM Forum Case Studies as a Basis for Coding and Analysis

The TM Forum has started publishing real-world solutions in a Case Study Handbook in May 2006 (TM Forum 2006a, b, 2007a, b, 2008a, b, 2009a, b). The cases have been submitted by telecommunication service providers, vendors or/and system integrators. There is no limitation of specific topics, as long as the case adheres to TM Forum standards (e.g. the "Solution Framework"). There was no interrelation between our research design and the publication of the cases. Therefore the cases can be seen as an unbiased basis for our research questions.

Each case has a one page summary, structured into header and content. The header contains information like title, company name and author. The content is structured into three sections: (1) business problem, (2) solution implementation, (3) deployment and result. We have considered the 231 cases published between May 2006 and December 2009.

We first selected all cases related to customer orientation, in a broad sense of the expression. For this, we conducted a full-text search over all cases. The search words are based on a recent publication of TM Forum about customer orientation (TM Forum 2009c): customer-centric, customer-centricity, customer experience, customer satisfaction, loyalty. We identified 38 cases, which are 14% of the total. For these cases we performed the coding processes described in Sect. 3.1. Figure 3 shows an example of the results of this coding process.

Case Title	Area	EA Layer	Aspect	Literally in Text
Quad-Play Network Management and Service Assurance Solution	problem	business processes	end-to-end	the end-to-end services and customers affected by network events and outages
	problem	information system	OSS	The challenge they were faced with was deploying a single Network Management and Service Assurance solution
	solution	business processes	operations	solution fit the eTOM model requirements for Operations–> Assurance–> Service Management & Operations and supported a standards-based focus
	solution	information system	web 2.0	provide a web-based solution (web 2.0) that would allow NOC staff, field operations and customers to visualize the impacted services and not just device alarms

Fig. 3 Example of coding results

4 Discussion of Results

Figure 4 shows the findings of our analysis and it follows a discussion of our questions.

Q1: Is customer orientation the sole driver of a transformation?

As we selected only cases related to customer orientation it is not surprising that the absolute majority (92%) describes it as a problem. The three remaining cases stress customer orientation as a benefit without mentioning it in the problem description.

Beside customer orientation, the projects are mainly motivated by monetary considerations, either revenue (61%) and/or costs (39%). For our question Q1 we looked into more detail in the 35 cases that have stated customer orientation as their initial trigger for the transformation project:

- "Customer Orientation" *alone*: six cases (17%)
- "Customer Orientation" *and* "Revenue": 14 cases (40%)
- "Customer Orientation" *and* "Costs": nine cases (26%)
- "Customer Orientation" *and* "Costs" *and* "Revenue": six cases (17%)

We see that the majority (83%) has combined customer orientation with revenue and/or costs. The clear answer on our first question is: transformation projects are driven by *several* quantitative and qualitative objectives, and not by customer orientation alone. This is in accordance with literature about the importance of assessing customer lifetime value and taking measures to maximize the revenue from *profitable* customers.

Most often, the improvement of customer orientation is not so much a strategic goal but rather a prominent factor, described in the context of information system

1. Problems ranked according to total number of appearance (N=38):

Area	EA Layer	Aspect	Amount	Percent
Problem	Strategy	Customer Orientation	35	92%
Problem	Strategy	Revenue	23	61%
Problem	Strategy	Costs	15	39%
Problem	Information System	BSS	11	29%
Problem	Information System	OSS	8	21%

2. Solutions ranked according to total number of appearance (N=38):

Area	EA Layer	Aspect	Amount	Percent
Solution	Business Processes	Operations	27	71%
Solution	Information System	BSS	18	47%
Solution	Information System	Interfaces	12	32%
Solution	Business Processes	Strategy, Infrastructure and Product	11	29%
Solution	Information System	OSS	10	26%

3. Involved layers per case (N=38):

EA Layer	Amount	Percent
Business Processes & Information System	24	63%
Information Systems only	8	21%
Business Processes only	6	16%
Total	*38*	*100%*

Fig. 4 Findings of our case study analysis

problems. In total, 25 cases (65%) mention at least one aspect related to the information system layer, while only 13 (34%) do so for the business process layer. It seems that the related information systems are more visible to the authors of the case study compared to the business processes. In some cases they also mix the problem description with the already known solution (e.g. they have solved the problem by implementing a new BSS, therefore, they describe the old BSS as the problem).

Q2: Which parts and aspects of an enterprise are changed to achieve customer orientation?

A clear majority (79%) of all cases describe the design of Operations Processes as part of the solution. At the next position we find Business Support Systems (BSS) (47%). Interfaces of information systems (32%), Strategy, Infrastructure and Product Processes (29%) and Operations Support Systems (OSS) (26%) are on the third to fifth position.

With regard to our question Q2, it becomes apparent that the majority of the cases describe Operations Processes and/or BSS as part of their solution (33 cases, 87%). They both appear under either the business processes' perspective or the information systems' perspective, and are linked to customer touch points, e.g., a process encompasses sales or a system supporting sales. But we have found that changes in *further* processes or systems have been deemed necessary. From an information system perspective the BSS must be linked to other systems (IT interfaces, 32%). For example, an efficient provisioning can only be assured by changes in the OSS (26%). From a business process perspective the launch of new products and technologies (Strategy, Infrastructure and Product, 29%) are also relevant. This shows that the improvement of customer orientation requires changes of different parts of the company. For example the case of an Indian mobile operator describes that market growth required a total change-over of business processes (Operations as well as Strategy, Infrastructure and Product) combined with BSS and OSS. The case of a service provider in Bosnia and Herzegovina states the clear need of an integrated order management system that was realized by a complete re-design of BSS and OSS combined with SOA technology and based on end-to-end processes.

Both layers "business processes" and "information systems" appear in 63% of the solutions, again answering Q2.

The cases that only describe information systems in their solution all mention information systems as part of the original problem. These cases refer to a technical solution, e.g. "Automating Exception Management and Service Fulfillment", "eService Kiosk". Business requirements and related changes of business processes were not part of the described projects. In all eight cases the described solution is a commercial-of-the-shelf system (COTS). For example, a case of a service provider in the Asia-Pacific region describes the implementation of a COTS product portfolio management system for "product-centric transformation".

The six cases that refer to business processes only cover a preliminary phase of the project, i.e. contain no implementation of a solution. For example the case of a mobile carrier in Kuwait describes the identification of critical or missing business processes.

5 Conclusion, Limitations and Outlook

With the increasing maturity of telecommunication markets, the role of customer orientation becomes more important to the market actors. Our study of real-life cases shows, though, that customer orientation mostly comes along with goals associated to cost and/or revenue. In most projects, telecommunication companies address both business processes and information systems. It turns that increasing customer orientation requires changes in and integration of customer- and production-related activities and systems of the company.

With regard to our expectation that customer orientation is a major concern to telecommunication companies, it is rather surprising that only 14% of all cases explicitly mention customer orientation aspects for their projects.

A screening of both the selected and the remaining cases shows two major limitations. Both the scope and the target audiences of the cases vary substantially: the cases vary from company-wide projects to detailed technical solutions; some technical cases do not show the overall context. In addition, our research is limited to the analysis and description of changes. The implementation of these changes requires consideration of further topics like IS development and planning of IS projects.

Furthermore, there is an *evolution of requirements and priorities over the last years*. The collection of cases started in 2006. From our experience, the telecommunication industry realized the importance of customer orientation during the last 2 years. Indeed, the majority of relevant cases appeared in 2009 (16 cases) and 2008 (10 cases), followed by five cases in 2007 and seven in 2006.

The presented work builds upon the analysis of approximately 15% of the case studies. In a next step, we will analyze the whole set of 231 cases. Hence we will also identify those cases that are dealing with customer orientation without mentioning it explicitly. The analysis of all cases will furthermore allow us to formulate research questions over a broader set of aspects relevant to transformation in the telecommunication sector, e.g. the role of systems integration or the aspect of product bundling.

References

Ahn JH, Skudlark A (2002) Managing risk in a new telecommunications service development process through a scenario approach. J Inf Technol 17(1):103–118

Bendapudi N, Berry LL (1997) Customers' motivations for maintaining relationships with service providers. J Retail 73(1):15–37

Bertin E, Crespi N (2008) Describing next generation communication services: a usage perspective. In: First European conference ServiceWave 2008, pp 86–97

Bolton RN, Lemon KN (1999) A dynamic model of customers' usage of services: usage as an antecedent and consequence of satisfaction. J Mark Res 36(May):171–186

Bruce G, Naughton B, Trew D, Parsons M, Robson P (2008) Streamlining the telco production line. J Telecommun Manag 1(1):15–32

Choi M, Won-Ki-Hong J (2007) Towards management of next generation networks. IEICE Trans Commun 90B(11):3004–3014

Czarnecki C, Heuser M, Spiliopoulou M (2009) How does the implementation of a next generation network influence a telecommunication company? In: Proceedings of the European and mediterranean conference on information systems (EMCIS), Turkey, 13–14 July
Fink L, Markovich S (2008) Generic verticalization strategies in enterprise system markets: an exploratory framework. J Inf Technol 23(4):281–296
Fornell C, Johnson MD, Anderson EW, Cha J, Bryant BE (1996) The American customer satisfaction index: nature, purpose, and findings. J Mark 60(Oct):7–18
Frank U (2002) Multi-perspective enterprise modeling (MEMO)—conceptual framework and modeling languages. In: Proceedings of the Hawaii international conference on system sciences
Grida I, Bertin E, Crespi N (2006) Next/new generation networks services and management. In: ICNS'06. International conference on networking and services. Slicon Valley, CA
Gustafsson A, Johnson MD, Roos I (2005) The effects of customer satisfaction, relationship commitment dimensions, and triggers on customer retention. J Mark 69(Oct):210–218
Johnson MD, Fornell C (1991) A framework for comparing customer satisfaction across individuals and product categories. J Econ Psychol 12(2):267–286
Knightson K, Morita N, Towle T (2005) NGN architecture: generic principles, functional architecture, and implementation. IEEE Commun Mag 43(10):49–56
Lee CS, Knight D (2005) Realization of next generation network. IEEE Commun Mag 43(10):34–41
Mikkonen K, Hallikas J, Pynnönen M (2008) Connecting customer requirements into the multi-play business model. J Telecommun Manag 1(2):177–188
Mittal V, Kamakura W (2001) Satisfaction, repurchase intent, and repurchase behavior: investigating the moderating effects of customer characteristics. J Mark Res 38(Feb):131–142
Österle H, Blessing D (2003) Business engineering model. In: Österle H, Blessing D (eds) Business engineering. Springer, Berlin, pp 65–85
Peppard J, Rylander A (2006) From value chain to value network: insights for mobile operators. Eur Manage J 24(2):128–141
Reilly JP, Creaner MJ (2005) NGOSS distilled. The essential guide to next generation telecoms management. TeleManagement Forum, Morristown
Schubert P, Williams SP (2009) Constructing a framework for investigating and visualizing ERP benefits and business change. In: 22nd Bled eConference eEnablement: facilitating an open, effective and representative eSociety. Slovenia, 14–17 June
TM Forum (2006a) Case study handbook, 35 real-world solutions using TM Forum Standards, vol 1. May 2006
TM Forum (2006b) Case study handbook, 32 real-world solutions Using TM Forum Standards, vol 2. Dec 2006
TM Forum (2007a) Case study handbook, 34 real-world solutions using TM Forum Standards, vol 3. May 2007
TM Forum (2007b) Case study handbook, 28 real-world solutions using TM Forum Standards, vol 4. Nov 2007
TM Forum (2008a) Case study handbook, 17 real-world solutions using TM Forum Standards, vol 5. May 2008
TM Forum (2008b) Case study handbook, 22 real-world solutions using TM Forum Standards, vol 6. Nov 2008
TM Forum (2009a) Case study handbook, 33 real-world service provider and vendor solutions using TM Forum Best Practices & Frameworks, vol 7. May 2009
TM Forum (2009b) Case study handbook, 30 real-world service provider and vendor solutions using TM Forum Best Practices & Frameworks, vol 8. Dec 2009
TM Forum (2009c) Customer experience management: driving loyalty and profitability. TM Forum Insights Reports
Urbaczewski L, Mrdalj S (2006) A comparison of enterprise architecture frameworks. Issues Inf Syst 7(2):18–23

Winter R, Fischer R (2007) Essential layers, artifacts, and dependencies of enterprise architecture. J Enterp Archit 3(2):7–18
Yin RK (2003) Case study research: design and methods. Sage, Thousand Oaks
Zachman JA (1997) Enterprise architecture: the issue of the century. Database Program Des 5(2): 1–13

Convolution as the Key for Service-Identification in Complex Process Models

Gunnar Dietz, Martin Juhrisch and Katrina Leyking

Abstract Process models contain a lot of information. However, even created with the goal of enhancing the understanding of complex processes by making them transparent in an easy human-readable form, it still remains to be a hard task to analyze them in an automated computer-based way. While service-oriented architectures offer a technical base for service-management, the question of mapping business requirements to a design that remains maintainable is still in discussion. In this paper a new approach to ease the analysis of complex models is presented. It heavily relies on a new convolution theory that allows to generate ratios, which can be used to determine the applicability of a service candidate for its use as a real service within a SOA.

1 Introduction

Business processes are now more than ever supported by information and communication technology (ICT) and simultaneously determined by this. On the one hand, ICT allows the implementation of business processes; on the other hand, its flexibility determines a company's ability to respond to changing process requirements. A discussion on the agility of an enterprise is therefore increasingly linked with a discussion of the flexibility of its ICT systems [3, p. 3, 30, p. 6, 32, p. 80]. In this context, the concept of service-oriented architecture (SOA) is thoroughly investigated in recent years.

Service-orientation is currently considered to be the most viable and important approach for the development of business application systems. The expectations

G. Dietz (✉)
Global Institute of Management and Economics, Dongbei University of Finance and Economics, Dalian 116025, China
e-mail: mail@gdietz.de

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_15,

to this approach—to close the semantic gap between business requirements and the technical implementation—are very high. Many system providers nowadays characterize their systems to be service-oriented.

With service-oriented architectures, companies are faced with a paradigm shift: In the future services play a key role as components of their enterprise architectures [6, p. 123, 7, p. 10, 23, p. 117, 28, p. 1]. The design of a service-oriented business application system aims at creating a distributedly implemented technological basis, which allows an efficient response to business needs. A service encapsulates a reusable amount of system functionality. The combination of services allows the automation of business processes within and across the boundaries of an individual enterprise [37, p. 141]. The compliance with appropriate service design criteria ensures that services can be orchestrated to so-called process-oriented service compositions.

The development of service compositions is associated to the challenge of mapping business requirements to a set of service candidates. In a top-down approach that means to identify processes that are suitable for an implementation as a Web service. Of special relevance is the identification of services as subsets of an amount of processes. There are approaches that help to identify services with the help of process models. However, these approaches fail in the case of a high number of distributedly developed process models with complex interdependencies.

In this scenario the task of identifying services remains a highly manual task and causes extensive effort. An increased efficiency could be achieved if the necessary tasks are solved in a completely model based way. Then algorithms, based on a set of formal rules, could perform model operations. Thus, the required time and effort for solving the overall task would be reduced. The detection of similarities and differences between process models is the prerequisite for an automated model comparison. In this paper an approach is presented that allows a model-based identification of SOA services. This approach relies on ratios that measure complexity, cohesion and coupling. Since these ratios often depend on the process description (which means how the process was modeled), we propose a method of determining such ratios in a process inherent way.

Our research goal is to establish a method for the identification of service candidates. The authors follow the paradigm of design research. Knowledge is gained through the creation and evaluation of artifacts in the form of languages, models, methods, or systems. In contrast to empirical research, design science aims not on the assessment of research findings with respect to their veracity, but on the usefulness of the constructed artifact in the sense of problem solving (see [17, p. 82; 25, p. 253].

The paper is organized as follows. In the next section we start with a theoretical background. Section 3 discusses related work. In Sect. 4 we identify requirements for the automatic analysis of process models. Finally, Sect. 5 introduces our solution—the Description Kit Approach. The paper ends with a discussion, summarizing the research results and exposing open questions regarding the identification of service candidates.

2 Design Principles

There are two possible methods for designing services: Top-down, from the level of the business process description to the service design, and bottom-up, from the level of existing application systems to the service implementation. In practice the bottom-up approach is commonly used, since that means to introduce a SOA on the foundation of existing application systems [2, 11, p. 36 and 366, 38, p. 567, 39, p. 35 and 366].

The top-down approach, on the other hand, focusses on identifying those processes that are applicable for an implementation as a service [11, p. 363, 35, p. 258].

In this article we assume the case that for an existing SOA there exist services, which were developed using both approaches. The main problem when designing services is a deficit of clear guidelines for the functional outline of a service. The basis for such a decision are the business processes that have to be supported, the business objects derived from these processes, as well as the importance of several design goals. Clear design criteria are evident for a systematic identification of service candidates on the basis of business process models [2, 12, p. 152].

Schwemm et al. derive five design criteria from literature: Business orientation, self-containedness, modularity, interface orientation and interoperability [30, p. 31].

The design principle of business orientation refers to the granularity of service functions. The granularity has to match the extent of functionality that is provided by the service function [15, p. 43 and 48]. A service is business oriented when it contains all objects that are necessary for a certain task [30, pp. 32]. Dependences between different services have to be analyzed. Service candidates are a single process or a set of processes that serve a common purpose and use the same business objects. A measurement for the coherence of a system is cohesion [26, p. 315].

The self-containedness (or autonomy) of a service highly influences its maintainability. Self-contained systems are better to maintain compared to dependent systems, since changes only marginally influence other systems [31, p. 467, 34, p. 1282]. Simon operationalizes self-containedness with the help of coupling [31, p. 467]. Coupling is a measurement for the pairwise dependency of subsystems [34, p. 1282]. A process can be identified as a service candidate if it is independent from other processes. That means that firstly it uses business objects that are not used by other processes, and secondly the business objects that are transferred to other processes are of small complexity [36, p. 108].

Modularity refers to the fact that services must be designed in a way that they can be used as a part of a complex business process [29, p. 414]. They should be interchangeable and reusable in different service compositions. From these facts some principles for the service design can be derived.

The design principles interface orientation and interoperability focus on the assumption that services should represent stable interfaces whose functionality and technical specification is completely described (using metadata) [30, p. 31].

To combine process functionality into services is an optimization problem. In this article we propose the determination of dependencies between different functions and information objects by “folding” or “convolving” business process models and by the generation of ratios based on the convolution result. Self-containedness and business orientation will be operationalized by cohesion and coupling. The ratios measure differences and similarities between different partitions of processes into services (better said service candidates) and their relationships. A set of processes is an ideal service candidate, if the services within the set have high cohesion, high coupling with each other and low coupling with processes not belonging to the set.

3 Related Work

Service identification, i.e. locating functions that appear worthwhile being exposed as service, is considered as critical part of service-oriented system engineering [5, 18, 22, 35]. It is one of the first conceptual activities to be conducted in a SOA development project. Quality of identified service candidates determines overall system quality to a large extent. Flaws in this activity propagate to all later activities causing cost-intensive iterations. Due to this high impact of service identification, systematic and thorough techniques are imperative [10, p. 80]. Heutschi [16, p. 155] differentiates between project-driven service identification and strategic service portfolio planning. The former is limited to case-based service requirements formulated by business departments and realized by the IT shop within a specific service-oriented software development project. This rather pragmatic, erratic service construction facilitates quick results and avoids additional efforts for architecture-wide process and domain analysis. However, these apparent benefits come with the cost of a domain-oriented, cross-project service design. Project focused service identification neglects architecture-wide service reuse and service improvements. On the other hand, long-term service portfolio planning requires a more founded approach that eventually allows for sourcing or commercialization strategies or capacity planning. This requires the central IT shop to cooperate with business domain owners. They jointly decide which services shall be provided for end-to-end business processes. Such a provider-oriented service identification enables SOA architects to assess and plan overall reuse potentials, capacity and resource requirements. Therefore, Heutschi and Dodd [9] recommend service portfolio planning specifically for domains that include functions of high strategic relevance and high reusability, outsourcing or commercialization potential.

Potential service functionality can be searched for in different contexts. Typically, one differentiates top-down service identification in business process structures [35] from bottom-up service identification in existing software systems [27]. However, this distinction is somewhat flawed as it does not separate the analysis of as-is situations from the analysis of to-be situations. As-is business processes and resources reveal service candidates based on the present status that incorporates a long history of experience and investments. Despite the relevance of accumulated process knowledge, limiting the analysis to the current business and IT-context would risk propagating current inefficiencies to the new system. Mitigating process weaknesses, to-be business process models represent an improved business context for service identification. On the IT side, beyond existing IT applications, services available on the market shall be scrutinized.

Birkmeier et al. [4] classify existing approaches for systematic service identification along a series of dimensions that include the underlying service notion (technical or business-driven), degree of formalization, availability of procedural guidelines, development direction (top-down, bottom-up, meet-in-the-middle), use of exact or heuristic optimization techniques, model views (data, functions, processes), consideration of existing system structures and system dependencies, distinction of service hierarchies and types, tool support, quality assessment and validation techniques (none, plausibility, application case, best practices). The authors classify thirteen service identification approaches which originate in component based software development or service-oriented system engineering. The results of this analysis show that none of the approaches fulfills all criteria. Notably, older component-based approaches use more structured, algorithm-based techniques and validation methods than more recent service-oriented approaches. Thus, the authors identify a need for more formal, detailed service identification techniques that may also include optimizing algorithms.

To identify service candidates within business process models more systematically, Leyking [24] proposes a framework of service eligibility indicators that assess a process activity's suitability for service support. She refers to and advances related approaches along five perspectives that add up to an overall eligibility indication. From a functional perspective, service granularity is defined by clustering algorithms as proposed by Aier [1]. Service reusability is addressed from a process oriented perspective by the number of occurrences of an activity across all processes [4, p. 66], as well as the number of invocations of an activity over the course of time [33, p. 90]. From an organizational perspective, Klose [22] differentiates between visibility and interaction indicators that attribute to the process participants' request and ability to participate in the activity. Such potentials for out tasking or opening the activity to process stakeholders originates in the service marketing literature (line of interaction and line of visibility) and is measured by binary values. From a business perspective, Levi and Arsanjani [2] present goal-service modeling (GSM) as an approach to prioritize service candidates according to their business relevance. This is complemented by the event-driven perspective proposed by Gold-Bernstein et al. [14] that takes into account whether a process activity responses to a non-trivial, significant business event. From a data perspective, Esswein et al. [13] propose an approach that measures the degree of coupling of functions by assessing its data exchange behavior. The fewer functions that use the same data objects and the less complex the shared data objects are, the looser the coupling of functions is [28, p. 418]. The complexity of shared data objects is determined by the number of mutual relationships as modeled in a data model. As a side effect, low complexity of input/output data relates to lower data volumes, as it is required for improved overall performance [22, p. 141]. A further indication about the autonomy of a function is given by the cohesion of the data objects used by the function. Esswein et al. [13] derive this figure of cohesion based on the number of mutual relationships between data objects. The underlying assumption is that a function can be considered cohesive if its data objects have few relationships with each other. Normalizing the number of relationships by subtraction from the number of data objects delivers a value that indicates the degree of cohesion.

4 Conflicts in Modeling and Guidelines

Previous approaches like the above mentioned Esswein approach often rely only on simple counting of objects. However, the result of such an analysis is highly dependent on how the model was created. Two descriptions of the same process may yield completely different results. This effect is even increased by the fact that the creation of models is often done independently of their later use in a semi-automatic analysis of them.

There are several possible conflicts that, when not handled properly, make a comparison of process models unfeasible. The most important are (see also [21]):

Language conflicts: Language conflicts affect the labels of model elements to compare. The avoidance or solution of homonyms and synonyms is the first prerequisite for model comparison.

Structure conflicts: Each modeler has the freedom to describe his domain at a specific level of abstraction and may choose a certain degree of detail. This leads to so-called structural conflicts.

Type conflicts: Type conflicts arise from choosing one of various choices of an appropriate grammatical concept for modeling.

Domain conflict: A domain conflict results from the desire to compare models out of different domains. The conflict is a consequence of a varying way of modeling in different domains or occurs because different kinds of artifacts are compared.

Many approaches try to resolve these conflicts with different methods (especially linguistic methods) after the process models have created. Even if these methods become more and more powerful and research goes on in this area, a comparison of process models is much easier, when the problem is attacked earlier. Clear guidelines for the modeling process clearly will result in models that are easier to compare. This is especially true, if the models should be used in an automated way. Juhrisch suggests that a restriction of the freedom of the modeling based on a certain understanding of guidelines could be useful [20]. Guidelines enhance the possibility to establish a traceable connection to the real world domain.

To implement guidelines in modeling, the so-called Description Kit Approach is introduced, which contains artifacts and a methodology for creating or enriching modeling languages in a "guided" way, as well as some algorithms, which automize generic comparison problems based on the fact that the now created models inherit a better comparability. This approach makes it possible to analyze complex process models with a newly introduced concept called convolution and to generate ratios based on this analysis. This should be described in the following sections.

5 The Description Kit Approach

The Description Kit Approach (DKA) offers a framework for describing guidelines and can be seen as an autonomic part within the model-driven method. It can be used in combination with other modeling methods or languages or on its own. It offers a generic way to create models, which should serve as a foundation to serve a certain problem in mind. The artifacts of the DKA are *Descriptions, Description Kits* and *Description Kit Type.* This multi-layered approach makes it possible to create or enrich a modeling language in a way that the modeler is lead through the modeling process by guidelines when using the language. A language created in this way is called a Description Kit Language. For details we refer to [21].

5.1 Description Kit Languages

The main idea behind a Description Kit Language (DKL) is to provide a framework for using so-called Descriptions (Descs). Model creation implies the creation of Descriptions, while the framework ensures that the modeler follows the inherent guidelines that are part of the DKL.

Description: This is a description of some facts within a framework of guidelines.

The concept of descriptions is a hierarchic concept, which means that one description may include other description. Therefore it is theoretically possible to create arbitrary complex descriptions.

To implement guidelines, a DKL must contain information about how to use certain descriptions. This is done by the concept of so-called Description Kits (DescKits or DKs). They supply the necessary framework for a certain set of descriptions in the sense of restricted modeling.

Description Kit: A description kit defines guidelines for the description of a concept.

However, the DKs are not the language, but merely represent the guidelines for language usage. Language creation, one the other hand, here the creation of a certain DKL, means to define "types" for modeling constructs. This is reflected by so-called Description Kit Types (DescKitTypes or DKTs).

Description Kit Type: A description kit type determines a concept that should undergo guidelines.

When compared to ordinary meta-modeling concepts, Descs and DescKitTypes represent language usage and language creation. The newly introduced concept of DescKits can be thought of some kind of "glue" between these two modeling phases, which makes it possible to introduce guidelines in a structered way. For details we refer to [8, 19].

5.2 Generic Mapping Algorithm

The DKA also contains some algorithms, which are formulated and implemented in a generic way. They have a high focus on model comparison, but keep generic applications for model-based problem-solving methodologies in mind. In fact, the key to solve problems in a model-based way is to ease an understanding of models, especially when thinking of an automated (machine-based) analysis of models. Understanding on the other hand in most cases means to compare to already understood concepts.

The starting point for the algorithms is a 1:1 comparison algorithm for two single descriptions. Since a description may contain other descriptions, there is already some work to do. On the other hand, the hierarchic structure of descriptions allows already to derive more information than by comparing two single ordinary modeling artifacts.

For details on the 1:1 algorithm we refer to [8].

The next step is to compare two complete models or parts of models. They main idea here is to "fold" a complex model along the relations into a single, but artificial, big description that contains all necessary information for the comparison. We denote this part of the algorithm as "convolution".

5.3 Convolution of Process Chains

Also the convolution algorithm is formulated generically. The convolution operates step-by-step on a model and "combines" descriptions along all relations until the model completely falls together into a single description. This means that we need a convolution operation for each relation type.

In the context of service identification this is done as follows: A possible DKL for service descriptions contains DescKitTypes "Interface", "Input", "Output", and "Object". A description of an interface therefore means to describe (among other things) which objects serve as input and as output, and includes a description of these objects. Each process activity is therefore enriched by such a description, while relations in a process model represent the sequence of processes and therefore the object flow along the process.

In this scenario the convolution operation can be described as follows:

$$\begin{aligned}&\Big(\big\{\text{Interface}\{\text{Input}\{\text{Object } I_1\}\}\{\text{Output}\{\text{Object } O_1\}\}\big\}\\&\qquad\xrightarrow{\text{Process Flow}}\big\{\text{Interface}\{\text{Input}\{\text{Object } I_2\}\}\{\text{Output}\{\text{Object } O_2\}\}\big\}\Big)\\&\quad\xrightarrow{\text{Conv. Result}}\big\{\text{Interface}\{\text{Input}\{I_1\cup(I_2-O_1)\}\}\{\text{Output}\{(O_1-I_2)\cup O_2\}\}\big\}\end{aligned}$$

Here "$\cup$" and "$-$" are two predefined operations, which represent two different meanings of "combination". Both rely on the 1:1-mapping algorithm to detect equality (or similarity) of objects: "∪" represents a union in a natural way, while "−" is an operation that removes temporary objects (intermediary results of one process activity that are consumed later on by another process activity and not used anymore later on) from the results.

5.4 Generation of Ratios

Using this convolution operation we are now able to convert model data into descriptions that serve as a good foundation to generate ratios. Since the convolution process considers the model structure and can detect object flows along relations, the resulting description contains a certain object only once, but still stores the information related to what happened to this object. Some details of the object may be used in the beginning phase of a process, others in a latter part, but the convolution result contains both parts. Therefore the convolution can be used to detect or measure coherence of a given process (or subprocess).

A highly coherent process would collapse during the convolution. All similar occurrences of (descriptions of) an object will be matched and collected to only one part in the resulting big convolution description. Therefore a measurement of coherence would be to compare the complexity of the original process and the complexity of the convoluted process. This could be used to analyze all business processes in search of service candidates. Similarly a measurement for the coupling of processes could be established, which will be done below.

First we need a measurement for complexity. The complexity of a (sub)process could best be derived from the objects, which means here in the case of the DKA their descriptions. The DKA with its embedded structures has the big advantage that complexity now can be derived not only by the number of objects, but a look into the objects can be done by analyzing the descriptions. Counting different description elements with different weights (depending on their type) yields a measurement for the complexity of a description. An easy approach to define the complexity of a (sub)process would be to add the complexity of all descriptions of a process and to add some value for different types of relations. A better measurement would include an analysis of the underlying graph of the relations: Are there loops, iterations, branches?

Nevertheless, complexity defined in that way is not an inherent measurement for a process, since it depends on how this process was modeled, especially on the granularity. Here another advantage of the DKA is that variance in granularity is limited because the inherent guidelines for (a restricted) modeling control this granularity. However, the complexity still remains a vague measurement. We denote the complexity of a (sub)process P by $K(P)$.

Now define the cohesion by

$$\Lambda(P) = \frac{K(P)}{K(F(P))}$$

Here $F(P)$ denotes the convolution result of P. This formula is exactly what was mentioned above: Compare the complexity of the original process with the folded one. By this construction now a variance in granularity cancels out. A courser description of P would result in higher $K(P)$, but also higher $K(F(P))$, so the coherence is independent of that. We get a measurement of the cohesion of a process that is an inherent value for the process.

The cohesion is a measurement of how well all functionalities that are part of a (sub)process fit together in the sense that they operate on information objects with high connection or context. The convolution operator given above has the property that objects with high context fall together while different objects still remain separated. Furthermore the convolution process is done along the relations, so that not only the objects (including their status) are considered, but also the structure of the process influences this measurement. The convolution highly relies on the 1:1 mapping algorithm described in [21].

The coupling on the other hand is a measurement that is in some sense working in the other direction compared to cohesion. Two subprocesses are coupled (have high coupling) if they—at least partly—operate on the same or similar objects. A measurement for this is the value

$$\bar{\kappa}(P_1, P_2) = \frac{K(F(P_1)) + K(F(P_2))}{K(F(P_1 \cup P_2))} - 1$$

(This formula uses the operator $\cup$ as described above.) This value is 1, if the convolution of the union of P_1 and P_2 has the same complexity as P_1 or P_2, respectively, which means the case of highest coupling, or 0, if the complexity of the union is just the sum of the complexities of P_1 and P_2, which means that both subprocesses have nothing in common and therefore are not coupled at all.

With the help of this value one can now define the coupling of a subprocess P compared to the complete process G as

$$\kappa(P) = \left(\sum_{P_j \in G \backslash P} \frac{K(F(P)) + K(F(P_i))}{K(F(P \cup P_i))} \right) - |G \backslash P|$$

This is the sum of the previous terms for all process steps P_i in G that are not part of P. However, this analysis can be time-consuming, because the number of submodels of a given model is exponential in the size of the model itself. Therefore it makes sense to use the described approach in a cascaded way: First fold small parts of the model, then fold some of the previous convolution products and so on. The modeler can support this procedure by pre-selecting some parts of the model.

6 Conclusion

This article continues some ideas that were created when developing the Description Kit Approach (DKA). It relies on the previously developed modeling approach and methodology that introduces guidelines for modeling and some algorithms, which had a focus on model comparison. These algorithms are used in this article to generate ratios (using a convolution approach) that can be used in service identification.

One of the advantages of the DKA is that it does not "reinvent the wheel". Descriptions can be used to enrich existing models, a re-creation of models using a different language is not necessary.

The usefulness of the DKA has been discussed in earlier publications. The generation of ratios however is an idea that lies beyond the initial purpose of the DKA and proves that the generic formulation of the DKA bears fruit. It is an exemplary demonstration of a reduction of a model-based methodology for solving a concrete problem (here service identification) to a model comparison scenario. Interesting is here, that comparison here does not (only) mean to compare two models, but (also) to compare different parts of *one* model.

The approach presented here helps to create ratios that are service inherent and to a high extend independent of the way a certain service was modeled. This allows a service identification that can be done semi-automatically, while the necessary human interaction is minimized.

References

1. Aier S (2006) How clustering enterprise architectures helps to design service oriented architectures. In: Proceedings of IEEE conference on service-oriented computing (SCC' 06), Los Alamitos, USA: IEEE Computer Society, pp 269–272
2. Arsanjani A (2004) Service-oriented modelling and architecture (SOMA). IBM developerWorks
3. Baskerville R et al (2005) Extensible architectures: the strategic value of service-oriented architecture in banking. In: Bartman D et al (eds) Proceedings of the 13th European conference on information systems, Regensburg, 2005
4. Bieberstein N et al (2008) Executing SOA: a practical guide for the service-oriented architecture. Pearson Education, Boston
5. Birkmeier D et al (2008) Zur systematischen Identifikation von Services: Kriterien, aktuelle Ansätze und Klassifikation. In: Loos P et al (eds) Modellierung betrieblicher Informationssysteme: Modellierung zwischen SOA und Compliance Management. Gesellschaft für Information, Bonn, pp 255–272
6. Brown AW (2000) Large-scale, component-based development. Prentice Hall, New Jersey
7. Cheesman J, Daniels J (2001) UML components: a simple process for specifying component based software. Addison-Wesley, Reading
8. Dietz G, Juhrisch M, Esswein W (2009) On the restriction of conceptual modeling—outlining an approach to enable business driven SOA. In: Proceedings of the 15th Americas conference on information systems (AMCIS), San Francisco
9. Dodd J (2005) Practical service specification and design part 2b: refining service requirements. CBDi J 2005:24–31
10. Endrei M, Ang J et al (2004) Patterns: service-oriented architecture and web services, IBM corporation
11. Erl T (2005) Service-oriented architecture: concepts, technology, and design. Prentice Hall, New Jersey
12. Erradi A, Anand S, Kulkarni N (2006) SOAF: an architectural framework for service definition and realization. In: Proceedings of the IEEE international conference on services computing, SCC2006. Chicago, pp 151–158

13. Esswein W, Weller J, Stark J, Juhrisch W (2009) Meet the challenge in service identification: a ratio-based approach. In: Proceedings of the 13th Pacific Asia Conference on Information Systems (PACIS), Hyderabad
14. Gold-Bernstein B, Ruh WA (2004) Enterprise integration: The essential guide to integration solutions. Addison-Wesley, Boston
15. Griffel F (1998) Componentware. Konzepte und Techniken eines Softwareparadigmas. dPunkt, Heidelberg
16. Heutschi R (2007) Serviceorientierte Architektur: Architekturprinzipien und Umsetzung in der Praxis. Springer, Berlin
17. Hevner AR et al (2004) Design science in information systems research. MIS Quart 28(1):75–105
18. Inaganti S, Behara GK (2007) Service identification: BPM and SOA handshake
19. Juhrisch M, Weller J (2008) Connecting business and IT—a model-driven web service based approach. In: Proceedings of the 12th Pacific Asia conference on information systems, Suzhou
20. Juhrisch M (2010) Richtlinien für die modellgetriebene Integration serviceorientierte Architekturen in Analysemodellen. Technische Universität Dresden, PhD-Thesis
21. Juhrisch M, Dietz G (2010) Constraints in conceptual modelling—outlining an approach to business driven web service composition. Int J Int Enter Manag 6(3):248–265
22. Klose K (2007) Service-orientierte Architekturen in der industriellen Auftragsabwicklung: Identifikation und fachkonzeptionelle Spezifikation von Services für ERP-/PPS-Systeme von Kleinserien- und Auftragsfertigern. Berlin: Logos Verlag—PhD Thesis, Westfälische Wilhelm University Münster
23. Kossmann D, Leymann F (2004) Web services. Informatik-Spektrum 27(2):117–128
24. Leyking K (2010) Service-oriented business process systems for knowledge work. Architecture, method, and application. Berlin, Logos Verlag, in print—PhD Thesis Saarland University, Germany
25. March ST, Smith GF (1995) Design and natural science research on information technology. Decis Supp Syst 15(4):251–266
26. McCabe TJ (1976) A complexity measure. IEEE Trans Soft Eng 2(4):308–320
27. Nadhan EG (2004) Seven steps to a service-oriented evolution. Bus Int 1:41–44
28. Papazoglou M et al (2006) Service-oriented computing: a research roadmap. In: Dagstuhl seminar proceedings 2006, Schloss Dagstuhl, Germany: Internationales Begegnungs- und Forschungszentrum für Informatik (IBF)
29. Richter JP, Haller H, Schrey P (2005) Serviceorientierte architektur. Inf Spektrum 28:413–416
30. Schwemm JW et al (2006) Serviceorientierte Architekturen: Einordnung im Business Engineering. Working Paper of the HSG, University of St. Gallen
31. Simon HA (1962) The architecture of complexity. Proc Am Philos Soc 106(6):467–482
32. Störrle H, Glock W (2008) Geschäftsprozessmodellierung für Service-orientierte Architekturen. In: Nissen V et al (eds) Service-orientierte Architekturen: Chancen und Herausforderungen bei der Flexibilisierung und Integration von Unternehmensprozessen. Gabler, Germany, pp 77–94
33. vom Brocke J (2008) Serviceorientierte Architekturen: Management und Controlling von Geschäftsprozessen. Vahlen—Habilitation, Westfälische Wilhelms-Universität Münster, Münster
34. Wand Y, Weber R (1990) An ontological model of an information system. IEEE Trans Software Eng 16(11):1282–1292
35. Winkler V (2007) Identifikation und Gestaltung von Services: Vorgehen und beispielhafte Anwendung im Finanzdienstleistungsbereich. Wirtschaftsinformatik 49(4):257–266
36. Yourdon E (1979) Structured design: fundamentals of a discipline of computer program and systems design. Prentice Hall, Upper Saddle River

37. Zeng L et al (2003) Flexible composition of enterprise web services. Elect Markets Int J Elect Comm Bus Media 13(2):141–152
38. Zimmermann O, Tomlinson M, Peuser S (2003) Perspectives on web services: applying SOAP, WSDL, and UDDI to real-world projects. Springer, Berlin
39. Zimmermann O, Krogdahl P, Gee C (2004) Elements of service-oriented analysis and design. IBM developer works: http://www.ibm.com/developerworks/webservices/library/ws-soad1/

Studying Maintainability on Model-Driven Web Methodologies

Francisco José Dominguez-Mayo, María José Escalona, Manuel Mejías and Jesús Torres

Abstract QuEF (Quality Evaluation Framework) is an environment to evaluate, through objective measures, the quality of Model-Driven Web Engineering (MDWE) methodologies. In this paper, this environment is presented and is used for the evaluation of the Maintainability in terms of various characteristics on MDWE. Given the high number of methodologies available and proposed over recent years, it has become necessary to define objective evaluation tools to enable organizations to improve their methodological environment and to help designers of web methodologies design new effective and efficient tools, processes and techniques and find out how it can be improved and how the quality improvement process could be optimized in order to reduce costs. This evaluation is applied to the NDT (Navigational Development Techniques) methodology, an approach that covers the complete life cycle and it is mainly oriented to the enterprise environment.

1 Introduction

Model-Driven Engineering (MDE) is a paradigm of software development which consists of the creation of models closer to a particular domain rather than concepts of a specific syntax. The domain environment specific to MDE for web engineering is called Model-Driven Web Engineering (MDWE) [9]. The Object Management Group (OMG) has developed the standard Model-Driven Architecture (MDA) which defines an architecture platform for proposals based on the Model-Driven paradigm. According to the OMG [15], the goals of MDA are portability, interoperability and

F. J. Dominguez-Mayo · M. J. Escalona · M. Mejías · J. Torres (✉)
Department of Computer Languages and Systems, University of Seville, Seville, Spain
e-mail: fjdominguez@us.es

M. J. Escalona
e-mail: mjescalona@us.es

M. Mejías
e-mail: risoto@us.es

J. Torres
e-mail: jtorres@us.es

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_16,

reusability through architectural separation. The concept of platform independence appears frequently in MDA. Models may have the quality of being independent from the characteristics of any technological platform. By applying this paradigm, the lifecycle of a software system is completely covered, starting from requirements capture, passing through the generation of code, and up to the system maintenance. In recent years, the growing interest in the internet has led to the generation of a high number of MDWE approaches which offer a frame of reference for the Web environment [8, 12]. On the other hand, there are a high number of approaches as OOHDM, UWE or WebML without standard consensus [16] a lack in the use of standards, and scarcity of both practical experience and tool support. An example of this methodology type is the NDT (Navigational Development Techniques) which is a methodological approach oriented towards Web Engineering. It is an approach defined in the Model-Driven paradigm and it offers a methodological environment. With the use of NDT-Suite, NDT offers tool support for each phase of the complete life cycle of a software project.

There are many methodological approaches in the area of MDWE and numerous comparative studies [9, 12]. Along these lines [16], must be considered, which specifically considers modelling concepts for their ubiquitous nature, together with an investigation of available support for Model-Driven Development in a comprehensive way, using a well-defined as well as fine-grained catalogue of more than 30 evaluation criteria. In [3], an approach is proposed for the evaluation of Web quality that provides all the elements which are in accordance with the ISO/IEC 14598. The idea of developing an MDE framework for evaluating quality has been applied in [14] and other papers of the same author. On the other hand, in the literature there are numerous references to metrics [2, 13], according to which, software measurement integration could be achieved by adopting the MDA approach. To this end, an approach is described in [10] for the management of measurement of software processes. From the methodological perspective, software measurement is supported by a wide variety of proposals, with the Goal Question Metric (GQM) method (Basili and Victor) and the ISO 15539 and IEEE 1061-1998 standards all deserving special attention. As for web metrics quality, in [4] some useful metrics proposed for web information systems are classified, with the aim of offering the user an overall view of the state of the research within this area. With regards to the metrics model, a significant study has been revealed in [1], which proposes a set of metrics for navigational models for the analysis of the web application quality in terms of size and structural complexity.

The term *quality model* is often used to refer to a set of quality attributes (also known as quality characteristics) and the relations between them. By answering "yes" and "no" to questions related to quality criteria, one may measure to what extent a quality criterion is achieved. ISO standards are set out in [11], where particular attention is paid to the ISO-9126 series with the hierarchical model of six quality factors and subcharacteristics related to each factor.

In the light of this situation, the need to assess the quality of existing methodologies arises. In this paper, QuEF (Quality Evaluation Framework), an environment for the quality evaluation of Model-Driven Web methodologies, is proposed.

The main goal of this research is to lay the basis of an environment for the assessment of MDWE methodologies that facilitates the quality evaluation of different methodological proposals under some objective criteria in order to improve these methodologies. Hence, there is a need for the suitable design of MDWE methodologies and effective tools. To this end, our work concentrates on evaluating and comparing existing proposals although the framework could be extended in the future to cover other areas. Furthermore, the software development process has a direct influence on the quality and cost of software development, and therefore the use of an MDWE methodology and its influence on the final product quality must be considered.

This paper is organized into the following sections. In Sect. 2 QuEF is defined, the stages for the definition of the Quality Model component for QuEF are given and descriptions of every component, structure, and the process required to achieve each component are outlined. Section 3 shows an application of QuEF to the NDT methodology. Finally, in Sect. 4, a set of conclusions and contributions is laid out, and possible future work is proposed.

2 QuEF (Quality Evaluation Framework) for MDWE

In this work, an approach, or *Methodology*, is a Model-Driven proposal for the development of web applications. It may provide a set of guidelines, techniques, processes and/or tools for the structuring of specifications, which are expressed as models. Only those web modelling approaches which are based on MDA in the framework are considered. In addition, a *framework* in this work is a basic conceptual structure composed of a set of elements used to evaluate, in this case, MDWE methodologies although it could be extended to cover other areas or domains. Therefore, an environment, QuEF, with a set of elements based on existing literature has already explained in other papers [5, 6], where four components for the evaluation of the quality of MDWE methodologies can be seen:

- *Quality Model component:* this includes the basis for the specification of quality requirements with the purpose of evaluating quality. It specifies each element and its purposes.
- *Thesaurus & Glossary component:* this includes all the necessary items to improve the standardization of the access channel and communication between users of different MDWE methodologies.
- *Approach Characteristics Template component:* this includes the description templates of the input methodology characteristics to be evaluated, and depends on the Quality Model description.
- *Quality Evaluation Process component:* this includes the definition and specification for the execution of the quality evaluation process.

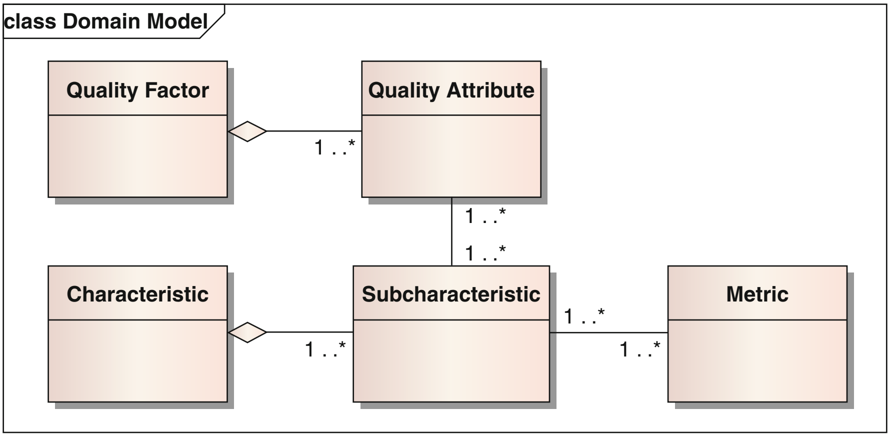


Fig. 1 Quality model metamodel

2.1 *The Quality Model Component*

The Quality Model in QuEF is a set of characteristics, subcharacteristics and metrics, quality factors, quality attributes and the relationships between these attributes, which provides the basis for the specification of quality requirements and the evaluation of quality in a specific domain (in this case, MDWE). In Fig. 1, the Quality Model metamodel with the relations between the several elements in the Quality model are shown, and the elements are described and explained.

- *Quality Factor:* This is a higher-level feature that affects the quality of an item. For example, a quality factor could be Usability, Functionality or Portability. Each quality factor and attribute in ISO 9126 is described in relation with a software product, however in our particular case all quality factors and attributes are described in relation with approach characteristics.
- *Quality Attribute:* A quality attribute is "A feature or characteristic that affects an item's quality (Syn: quality factor). In a hierarchy of quality attributes, higher-level attributes may be called quality factors, lower-level attributes called quality attributes". For example, Usability is defined for various quality attributes, such as Learnability, Understandability, and Operability.
- *Characteristic:* This is a higher-level concept of an approach. It may be, for example, the software development process, models, metamodels, languages, tools, transformations or the quality assurance techniques.
- *Subcharacteristic:* This is a lower-level concept of an approach. For example, the Model-Driven Engineering characteristic may have various subcharacteristics, such as the Language Definition, Transformations, and Trace Generation.
- *Metric:* In the Quality Model, metrics should indicate the degree to which a subcharacteristic is measured. In simple terms, a metric is used for measuring

subcharacteristics. For example, the evaluation may involve measuring quantitatively by means of metrics, or may use subjective evaluation in the form of inspections using checklists or interviewing the users. In terms of metrics, our aim is to look for a series of qualitative and quantitative metrics based on their nature, although it might be interesting to establish standard metrics on MDWE which are all, somehow, centralized. In the literature, numerous references to metrics can be found, but standardization has yet to be carried out. Furthermore, the metrics used must be validated theoretically or empirically.

In order to define a *Quality Model*, it must contain *association links* between the *subcharacteristics* and the *quality attributes.* These *association links* represent the dependencies between *subcharacteristics* and *quality attributes*. They show quality attributes which are affected by *subcharacteristics* or the areas of the methodology that will be significantly affected if the approach is changed. *Association links* may be based on proven and/or real-world experience. The impact of each *subcharacteristic* on *quality attributes* must be demonstrated and the requirements must be determined by real case study applications to a number of real projects. This should be supplemented by reference to published literature. Furthermore, *subcharacteristics* have to define quantitative or qualitative *metrics* which may be used to measure each *subcharacteristic*. Otherwise it would be necessary to define a set of indicators from reference values which may be set to a prescribed state based on the results of measuring or on the occurrence of a specified condition. Hence, a quality factor has various quality attributes and a characteristic has various subcharacteristics, as is shown in Fig. 1. A weight is used to define the importance of a metric in the value of a subcharacteristic. Similarly, a weight is also used to define the importance of a quality attribute in the value of a quality factor and the importance of the influence in association links between subcharacteristics and quality attributes. The tasks for the definition of the Quality Model, which have already been described in other papers, are:

2.1.1 Identifying Quality Factors

A set of quality factors based on current literature, such as ISO/IEC 9126, IEEE, and other standards which are adapted to MDWE methodologies, are identified, classified and placed in a hierarchy. The Quality Factors of an approach include Usability, Functionality, Reliability, Maintainability, and Portability. Each quality factor and attribute in ISO 9126 is described in relation with a software product, whereas in our study all quality factors and attributes would be described in relation with approach characteristics. In this work, *Maintainability* is taken as an example of the quality factor. In ISO 9126, *Maintainability* is a quality factor which is defined as: "*A set of attributes that bear on the effort needed to make specified modifications*". This definition could be adapted to more closely fit our specific domain: "*The ease with which a characteristic approach can be modified in order to: correct defects, meet new requirements, make future maintenance easier, or cope with a changed* *environment; these activities are known as methodology maintenance*" or in a general way could be described as: "*A set of attributes that bear on the effort needed to make specified modifications. The ease with which an approach characteristic can be modified to correct defects, modified to meet new requirements, modified to make future maintenance easier, or adapted to a changed environment.*"

2.1.2 Identifying Quality Attributes for Each Quality Factor

For each quality factor, a set of quality attributes has to be identified. For example, quality attributes related with *Maintainability* are described in the same way by adapting other definitions from ISO, IEEE, other standards and work already published. These quality attributes may be described as:

- *Stability:* The capability of a characteristic approach to avoid unexpected effects from modifications in the approach [ISO 9126].
- *Analyzability*: The capability of a characteristic approach to be diagnosed as having deficiencies or causes of failures in the approach, or the capability of identifying those parts yet to be modified [ISO 9126].
- *Changeability*: The capability of a characteristic approach to enable specified modifications to be implemented [ISO 9126].
- *Testability*: The capability of a characteristic approach to enable a modified approach to be tested [ISO 9126].

2.1.3 Identifying Characteristics

In MDWE, models are progressively refined and transformed into new models or code. To this end, tools may also be used to test, verify or validate the models. Moreover, each methodology may define its development process and/or techniques. The quality of methodologies in turn depends on the diverse characteristics, such as the *Model-Driven Engineering*, the *knowledge of MDWE methodology users*, the *web modelling*, the *customization modelling*, the *maturity* of a methodology, the *tool support,* and the *quality assurance techniques* applied to discover faults or weaknesses. The principal idea is to characterize the whole MDWE process.

2.1.4 Identifying Subcharacteristics and Metrics for Each Characteristic

Evaluating the degree to which the quality attributes would be affected is not an easy task, and for this reason most of the metrics defined so far are qualitative metrics which indicate if the subcharacteristic is Supported (S), Partly Supported (PS) or Not Supported (NS).

Table 1 Matrix of influences between subcharacteristics and quality attributes

		Maturity characteristic					MDE characteristic					Web Modelling characteristic						
		Topicality	Modelling Examples	Application in Real-World Projects	Publications	External Web References	Levels of Abstraction	Standard Definition	Model-Based Testing	Transformations	Traces	Web Conceptual Levels	Interfaces	Development Process	Content Feature Modelling	Presentation Feature Modelling	Navigation Feature Modelling	Business Feature Modelling
Maintainability quality factor	Stability	X	X	X	X	X		X		X	X		X	X				
	Analyzability	X	X	X	X	X	X	X			X	X						
	Changeability	X	X	X	X	X	X	X				X		X	X	X	X	X
	Testability	X	X	X	X	X			X		X		X					

2.1.5 Proposing a Set of Hypotheses for Linking Subcharacteristics to Quality Attributes

In this step, the association links between subcharacteristics and quality attributes are defined. A set of hypotheses are proposed to indicate which quality attribute is affected by each subcharacteristic. For example, Maintainability is described as a set of quality attributes. These quality attributes could be affected by one of various subcharacteristics as shown in Table 1. A first hypotheses is proposed in the table for the influences between subcharacteristics and quality attributes and they are showed for Maturity, MDE and Web Modelling characteristic and the Maintainability quality factor. This matrix could be made by expert designers and users and it has to be validated. The fuzzy logic is currently being considered in order to achieve real values, objectives and agreed.

2.2 The Other Components of the Framework

Other important element for QuEF is the *Thesaurus & Glossary* component. A thesaurus is a list containing the "terms" used to represent concepts, themes or contents of documents in order to make a terminological standardization to improve the access channel and communication between users of different MDWE methodologies.

We consider it necessary to carry out a standardization of terminology to improve the access channel for communication on MDWE. A set of concepts for MDWE methodologies is currently being described and related.

The Templates in the *Approach Characteristic Template component* with subcharacteristics and metrics for each characteristic are based on the Quality Model which is used to describe an input methodology. These templates would be used as input to QuEF. They would be analyzed in the evaluation process and compared with the model quality of the Quality Model component. Templates for MDE, Web Modelling, Tool Support, and Maturity have already been developed. In this work, Maintainability is studied in terms MDE, Web Modelling, and Maturity of the NDT methodology.

Finally, the *Quality Evaluation Process component* contrasts the information from each input approach template with information from the Quality Model. The main evaluation purpose is to identify tradeoffs and sensitivity points of the methodology under study. The idea is to determine which aspect needs to be improved on MDWE methodology. A simple evaluation is made using MS Excel which considers weights for metrics, subcharacteristics and quality attributes, although fuzzy logic is currently being considered in order to improve the evaluation process.

3 An Application of QuEF to the NDT Methodology

DT (Navigational Development Techniques) is an approach focused on the Model-Driven paradigm. Initially, NDT was focused on the requirements and analysis phases and it only defined a set of metamodels for the requirements and analysis phases and a set of transformation rules that let derive analysis models from requirements ones. Nowadays, it covers the complete life cycle of software development. In [8], some details about this evolution can be found. This evolution, some set of profiles let the use of UML tool based. Specifically, NDT-Suite uses these profiles to adapt Enterprise Architect [7] to use NDT. Besides, using the power of fusion and conexion of metamodels, NDT was extended to be adapted into enterprise environment. Nowadays, NDT is being used in several real projects and it is being involved in several new aspects like early testing or software quality.

3.1 Applying Templates to the NDT Methodology for the MDE, Web Modelling, and Maturity Characteristics

The Approach Characteristics Template component has been applied using an implementation in Microsoft Excel. However, the Approach Characteristics Template component has not yet been fully developed, and only the Tool Support characteristic, MDE characteristic, Web Modelling, and Maturity characteristics can be considered. In this example, the Transformations subcharacteristic is shown in Table 2 as an example of a template of a subcharacteristic and metrics. Qualitative metrics indicate

Table 2 Transformation subcharacteristic and metric

Model-Driven Reverse Engineering or Synchronization			
This uses standard languages for defining synchronization methods or reverse engineering techniques such as ADM, XMI, MOF; GXL, JMI, EMF, MDR, QVT.	NS		
It supports a Reverse Engineering Tool: A tool intended to transform particular legacy or information artifact portfolios into fully fledged models.			
Model 2 Model	NS		
Code 2 Model	NS		
It provides a synchronization method or a reverse engineering technique between transformations such as:			
PIM2CIM	S	Code2CIM	NS
PSM2PIM	NS	Code2PIM	NS
PSM2CIM	NS	Code2PSM	S

if the subcharacteristic is Supported (S), Partly Supported (PS) or Not Supported (NS). The total value for the quality attribute is the number of values divided by the total metrics in the subcharacteristic. The metric value in the example is 1 if it is *supported,* 1/2 of the arithmetic mean of supported elements from among the total elements (for example in transformations) if it is *partly supported,* and 0 if it is *not supported*. When these metrics are quantitative, an average value is taken, while an expected value is set as an ideal value for comparison with the values in the results. Although none of the subcharacteristics of the MDE, the Web Modelling, and the Maturity is shown, they are considered in the evaluation process of this example in the following section.

For example, total metric values of MDE subcharacteristics are shown in Fig. 2. In the figure, black bars represent NDT metric values for each subcharacteristic of the MDE characteristic respectively and grey bars represent the expected values for an ideal approach. In the figure is seen that the Standard definition, Model-Based Testing and Transformation subcharacteristics may be improved for the NDT methodology. On the other hand, the NDT methodology has a good score in Levels of Abstraction and Traces. It means that new trends or other subcharacteristics have to be added in templates in order to perform a quality continuous improvement of quality.

3.2 An Evaluation of the Maintainability on NDT Methodology

In the implementation in Microsoft Excel, Functionality, Reliability, Portability, Usability and Maintainability quality factors have been studied. In this example, quality attributes of Maintainability are shown together with their relations with the MDE, Web Modelling, and Maturity characteristics. This is shown in Fig. 3,

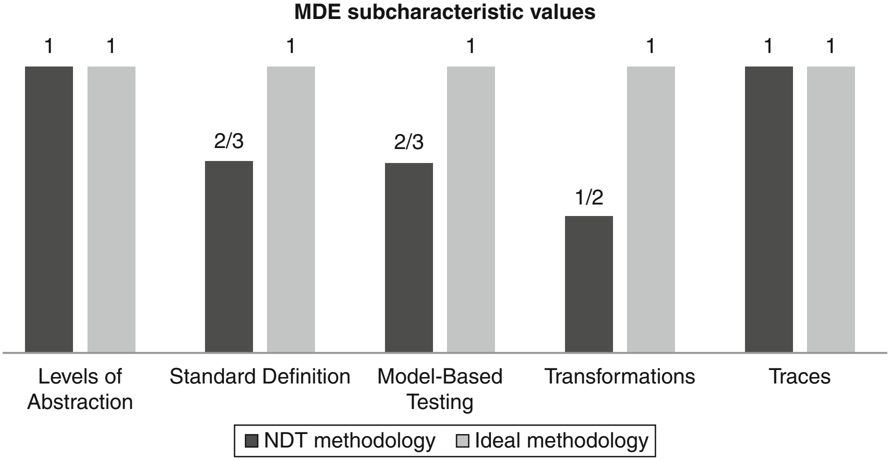


Fig. 2 Graph which represent maturity subcharacteristic values for NDT methodology

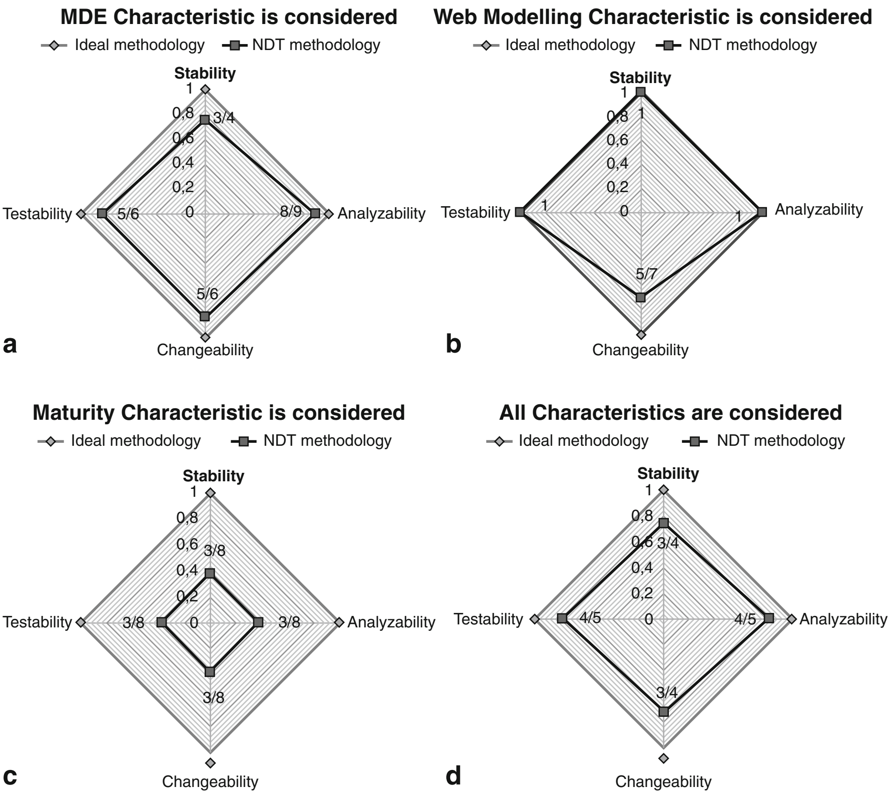


Fig. 3 Influences of several characteristics in the maintainability quality factor of the NDT methodology

where the black line represents Maintainability on the NDT methodology and the grey line represents the ideal Maintainability in an ideal approach, depending on the subcharacteristics under consideration. According to the results of the evaluation of the NDT methodology, only the MDE characteristic is considered in Diagram A, only Web Modelling is considered in Diagram B, and only Maturity is considered in Diagram C, for the evaluation of Maintainability. If all three characteristics are considered simultaneously then the results can differ greatly, as shown in Diagram D.

For the MDE characteristic, NDT has to improve the Stability quality attribute, which means that it would be difficult to avoid unexpected effects from modifications in the approach. For the Web Modelling characteristic, NDT has to improve the Changeability although it yields good results in Stability, Testability and Analyzability. In this graph we can see that results are uniform for this set of quality attributes (Testability, Stability and Analyzability), which could be due to: the similarity in the results; or the necessity to have more subcharacteristics and metrics for the identification of differences between these quality attributes, whereby characteristic templates would have to be defined in greater descriptive detail. In the Maturity graph, it can be observed that Maturity improves and renders the results more uniform. In general terms where all characteristics are considered, (Diagram D) NDT yields good results in Analyzability and Testability but has to improve the Stability and Changeability.

4 Conclusions

A framework is needed for the improvement of current proposals, and would be highly useful for the successful development of a new MDWE methodology. Therefore a quality environment for the assessment of MDWE methodologies is proposed. We consider that the use of QuEF will enhance the quality of products, processes and techniques of approaches. Furthermore, QuEF could be used for the optimization of a continuous improvement in quality since the number of sub-characteristics selected can be reduced, by using the matrix of influences, to include only those with the major influence in quality attributes. In previous papers [5, 6], we evaluated subcharacteristics related with MDE, Maturity and Tool Support of the NDT Methodology which are required for the measurement of the value of MDWE methodologies so that they can be assessed and improved in terms of their Functionality, Usability, Portability or Reliability. Therefore the use of QuEF can improve the efficiency and effectiveness of MDWE methodologies and in turn may lead to their more widespread use since this evaluation approach helps one understand the strengths and weaknesses of a methodology. This environment could be extended to involve other areas or domains. Further characteristics and quality factors have yet to be developed. To this end, Microsoft Excel is employed as a first prototype, although a software tool remains to be developed. Other methodologies, such as WebML and UWE [12, 16] are currently being evaluated in order to discover how they may be improved.

Acknowledgements This research has been supported by the project QSimTest (TIN2007–67843-C06_03) and by the Tempros project of the Ministry of Education and Science (TIN2010–20057-C03–02), Spain.

References

1. Abrahão S, Condori-Fernàndez N, Olsina L, Pastor O (2003) Defining and validating metrics for navigational models. IEEE computer society. In: Proceedings of the 9th international software metrics symposium (METRICS'03), Australia, p 200
2. Briand LC, Melo WL, Wüst J (2006) Assessing the applicability of fault-proneness models across object-oriented software projects. ISERN Report No. ISERN-00-06, Version 2, pp 1–20
3. Cachero C, Calero C, Marhuenda Y (2008) A quality-aware engineering process for web applications. In: Handbook of research on web information systems quality. Information Science Publishing, Hershey
4. Calero C, Ruiz J, Piattini M (2005) Classifying web metrics using the web quality model. Online Inf Rev 29(3):227–248
5. Domínguez-Mayo FJ, Escalona MJ, Mejías M, Ramos I, Fernàndez L (2010) A quality evaluation framework for MDWE methodologies. In: Proceedings of the 18th international conference on software quality management (SQM 2010), vol 1, No. 1, pp 171–184
6. Domínguez-Mayo FJ, Escalona MJ, Mejías M, Torres AH (2010) Towards a quality evaluation framework for model-driven web engineering methodologies. 6th international conference on web information systems and technologies (WEBIST 2010), vol 2, pp 191–194
7. Enterprise Architect. http://www.sparxsystems.com/
8. Escalona MJ, Aragón G (2008) NDT. A model-driven approach for web requirements. IEEE Trans Softw Eng 34(3):377–390
9. Escalona MJ, Koch N (2004) Requirements engineering for web applications—a comparative study. J Web Eng 2(3):193–212
10. García F, Serrano M, Cruz-Lemus J, Ruiz F, Piattini M (2007) Managing software process measurement: a metamodel-based approach. Inf Sci 177(12):2570–2586
11. ISO—International Organization for Standardization, ISO/IEC 9126–1. http://www.iso.org
12. Kroiß C, Koch N (2008) UWE metamodel and profile, user guide and reference. Technical Report 0802. Programming and Software Engineering Unit (PST), Institute for Informatics. Ludwig-Maximilians-Universität München, Germany
13. Lee Y-S, Liang B-S, Wu S-F, Wang F-J (1995) Measuring the coupling and cohesion of an object-oriented program based on information flow. In: Proceedings of the international conference on software quality. Maribor, Slovenia
14. Mohagheghi P, Aagedal J (2007) Evaluating quality in model-driven engineering. In: International workshop on modeling in software engineering (MISE'07). IEEE Computer Society, Washington, DC
15. OMG: MDA. http://www.omg.org/mda/faq_mda.htm
16. Schwinger W, Retschitzegger W, Schauerhuber A, Kappel G, Wimmer M, Pröll B, Cachero Castro C, Casteleyn S, De Troyer O, Fraternali P, Garrigos I, Garzotto F, Ginige A, Houben G-J, Koch N, Moreno N, Pastor O, Paolini P, Pelechano Ferragud V, Rossi G, Schwabe D, Tisi M, Vallecillo A, van der Sluijs K, Zhang G (2008) A survey on web modeling approaches for ubiquitous web applications. Int J Web Inf Syst 4(3):234–305

Using Antenarrative Approaches to Investigate the Perceptions of Information Systems' Actors Regarding Failure and Success

Lynette Drevin and Darren Dalcher

Abstract Many Information Systems are viewed as underperforming by failing to meet expectations or deliver value to customers. In order to try and understand why these problems occur we apply narrative methods in the IS domain, taking into account the stories that different groups of actors tell regarding their experiences of the development and use of new information systems. This paper presents the perspectives of multiple actors regarding success/failure and problems encountered in developing and using information systems from their experiences. Qualitative approaches are followed to gather, analyze and interpret the rich, multi-voiced and incoherent generated stories of stakeholders involved in software systems. It is shown that the antenarrative approach that is employed in this study can produce deeper insights into the experience world of involved actors.

1 Introduction

Success in Information Systems (IS) is very often defined in terms of cost, time and requirements criteria. However, looking from this perspective and interpreting media reports on this matter—the software developing industry is not ranked very successful (see for example: Standish 1999, 2001, 2004; KPMG 2005). Sometimes the literature also distinguishes between failed systems that are cancelled prior to installation and challenged systems that exceed their cost and schedule timelines and experience some trouble during development before eventually being put into operation. It is often difficult to define success when developing new IS. Different actors (stakeholders) are involved when systems are developed and put into operation. Therefore there will inevitably be different views on what is deemed a successful IS.

L. Drevin · D. Dalcher (✉)
Private Bay X6001, North-West University, Potchefstroom, South Africa, 2520
e-mail: Lynette.Drevin@nwu.ac.za

D. Dalcher
National Center for Project Management, Middlesex University,
Trent Park, London, N14 4YZ, UK
e-mail: D.Dalcher@mdx.ac.uk

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_17,

There are references in the literature that suggest rethinking the views of what success entails and using more levels of definitions of success. Dalcher (2009) for example refers to project management success, project success, business success and future potential. The idea of using the perceptions of multiple stakeholders when defining success is presented by Jiang et al. (2002) where both IS staff and IS users' view are taken into account in the study. The results show that differences exist in the two groups' views. Egorova et al. (2009) used three groups of users in a study and the stakeholders also indicated different factors that contribute to successful systems. The three groups were strategic, tactical and operational users and their differing perspectives on success were investigated. Most of the comparative studies are quantitative by nature utilizing questionnaires followed by detailed statistical analysis to obtain the results. These studies confirm the need for detailed investigation to seek out the real reasons for failure and success in order to become more successful in developing new systems. It is noteworthy that similar concerns have been expressed by researchers looking at failure phenomena for the last 35 years, suggesting that new approaches for evaluating the failure phenomena are still needed.

One of the approaches suggested for the understanding of IS failure situations is the interpretive approach, derived from the social sciences (Fortune and Peters 2005). The foundation is that reality is socially constructed and reality could be understood through the interpretation of data. Dalcher (2004) recommends using narrative methods where a failed or challenged IS situation can be seen as a special example of a case study, using the term "case history". The case history contains descriptions and observations of the actors and their perceptions and feelings, and thus will probably have a biased perspective. The failure situations, or negative experiences pertaining to a particular IS, can then be described via the unfolding stories of the actors.

Boje (2001) liberates a story from having a beginning, a middle, an end and a plot (Harmer and Pauleen 2007). Antenarrative is seen as "before story" or pre-narrative. Most of the stories of the actors involved in an IS failure situation will be of such a nature, that is to say incoherent, fragmented and non-linear.

The aim of this paper is to present findings of a study where antenarrative approaches are applied when the experiences of actors involved in the development and use of an IS are investigated concentrating on their perceptions of success and failure positions, as well as problematic situations.

The layout of the paper is as follows: In Sect. 2, some terms that are used are described. Section 3 give the background of the study and Sect. 4 highlights some methodological issues. The results are shown in Sect. 5 by presenting an analysis and discussion of the findings. The story is finalised in Sect. 6, with some concluding remarks highlighting the value of antenarrative approaches used.

2 Terminology

Usually many actors are involved when an IS is developed and additional ones join in when the system is put into use. Each actor has a role to fulfill; therefore everyone will have their own story to tell about their expectations, experiences and perceptions pertaining to a given system. An *actor* refers to any person who plays a role or participates during the development and/or use of an IS.

The Standish Group narrowly defines *successful* projects as completed on time and within budget with all features and functionality as originally specified. *Failed* projects are cancelled before completion (Standish 1999, 2001, 2003, 2004).

A few examples of failures are listed as follows: to show the extent and pervasiveness of the problem (Anon. 2007; Lyhne 2007; Telegraph 2007):

- The State of Florida welfare system: \$ 260 million in overpayments due to computational errors.
- The Seasprite helicopter program for the Australian navy: \$ 1 billion spent on helicopters that were grounded due to software problems.
- The FBI Virtual Case File, initial cost of \$ 170 million, declared a failure: Numerous requirement changes cost another \$ 104 million.
- eNatis: The South African Electronic National Traffic Information System cost ZAR 400 million. The list of problems are numerous and include: inadequate testing, project management "failures", security problems, roll-back strategy not in place etc.
- NPfIT: National Programme for IT, the UK's troubled ≤ 12 billion national, centralised IT programme which is currently being scaled back and facing the possibility of being formally withdrawn after the next elections.

Challenged systems are ultimately used after many changes and improvements have taken place. It could be that some of the actors are satisfied with the system while others are not. It could be that certain aspects of the system are accepted by some actors and other aspects are not. The term "failed situations" is used in this paper to refer to situations where there is a significant deviation from the expectations of the actors.

Lieblich et al. (1998) state that humans are storytellers by nature and that, through stories, researchers can explore and understand the inner world of individuals. Clandinin and Connelly (2000) see *narrative* as a way of representing and understanding experiences.

Boje (2001) introduced the term "antenarrative methods" to take into account that stories appear to be told improperly: in a fragmented, multi-plotted, and complex manner, e.g. a story in an *ante* state of affairs before a constructed and ordered narrative is used to impose sense.

3 Background

An academic environment was chosen for this research project due to the accessibility of the information system and the actors. Initial discussions with the IT services director and further meetings on other formalities (such as taking into consideration the relevant ethical issues) resulted in the permission for progressing with the project being granted. During initial meetings with IT personnel the outlay and infrastructure of the different IS were discussed. It was then decided to study a certain complex IS having around 19 subsystems. These include systems dealing with student records, grade processing, student fees, accommodation, applications and admissions, etc.

The system studied is contained in an environment called "Student Administrative Systems". The system is now in use although it is continually being updated and maintained to deliver the required functionality. For the study we concentrated on three subsystems, namely grade processing, student fees and student records. These subsystems are oriented more towards stakeholder-interaction thereby it was possible to get more meaningful feedback from the involved actors. During these meetings, individuals (actors) were identified who could provide us with their experiences on the specific IS. Four distinct actor groups were identified and it was decided to involve actors from each of the groups to gain access to the different sets of expectations and perceptions.

A qualitative approach was followed in order to get a diverse range of perspectives from the actors. Personal interviews were scheduled. A first meeting was held with one actor as part of a pilot study. The unstructured interview lasted approximately one hour. The interview was transcribed and analyzed following the protocols described by Lieblich et al. (1998): The whole interview was investigated and then sections were interpreted in context with the rest of the narrative to get a feel for the data. Lieblich et al. propose a model to classify types of narrative analysis in four categories: holistic, categorical, content-based, and the form-based approaches. The four approaches were evaluated for appropriateness in this study, and it was decided to use the first approach for initial analysis, (Drevin and Dalcher 2011).

After the pilot study was interpreted and the results reflected upon, it was decided to use a few approaches for analyzing the data, as there is no single best way to conduct narrative analysis and interpretation (Mishler 1995; Czarniawska 1998; Cortazzi 1993; Riessman 1993). The three-dimensional narrative inquiry space of Clandinin and Connelly (2000) was chosen in order to get a comprehensive view of each actor's experiences with the IS under investigation. These include aspects such as time and continuity (past, present, future), situation (place) and interaction (personal and social). However, these aspects were only used for data gathering and not during analysis as the reporting phase did not make use of these dimensions as such. Another approach that was selected was the antenarrative notion (Boje 2001), which is the theme of this paper. When the narrative approaches were investigated, antenarrative seemed to be an applicable way of studying the stories of the involved actors, as incoherency and multi-plotted narratives were evident.

Following the pilot stage, the meetings with the other identified actors were scheduled and interviews were held with 10 actors in total. Each interview lasted between 1 and 2 h, during which the actor shared his/her experiences with the specific IS with the researcher. Recordings were made of each interview and the data was transcribed in text format.

The next section describes in short the research methodology with the extended focus on the narrative perspective as followed for this project.

4 Narrative as Methodology

The research followed a qualitative approach. Two guidelines were taken into account during this project. Firstly, the view of Clandinin and Connelly (2000) that narrative inquiry studies the experiences of people was used in guiding the planning and conduct of the interviews with the relevant actors in order to encourage them to share their experiences related to the development and use of the IS. Secondly, the perspective introduced by Boje (2001) that antenarrative is the multi-voiced, non-linear and unplotted storytelling was also utilized in uncovering and making sense of actors' experiences. This guideline was useful in our study because the "stories" of the actors involved in the IS are accounts of events before plots or coherence are added to the storylines.

The current study borrowed from the narrative domain and imported new methods to the field of IS research. The experiences of actors involved with the IS were retold as their accounts and narrative methods could then be used as an adequate way of representing and understanding their experiences.

When the different narrative approaches were investigated for suitability, while acknowledging that there is no "single recipe", we found that the approaches that might be considered for the study of IS experiences had to take into account the fact that stories of actors involved in IS experiences are incoherent and unplotted. The selected approach had to look deeper, read between the lines, and look at the little stories, dualities, feelings and bias. By using such approaches, the researcher is placed in a position to generate a more comprehensive description and explanation of events, especially in those situations that were not acceptable or agreeable to the actors involved in the IS. The antenarrative approaches were selected for this study as they concentrated on the following applicable themes:

- Deconstruction: Attention was given to aspects such as the following: looking for dualities, the other side of the story, what has not been said, and opposites and other voices.
- Story network analysis: Here the focus was on identifying the connections between stories.
- Microstoria analysis: This goes hand in hand with the previous two approaches, where not only the more "official" standpoint is evaluated, but each actor's story is seen as important in order to get the complete picture of the phenomenon under investigation.

The next section describes the application of the antenarrative approaches to an IS as experienced and reported by the involved actors.

5 Analysis and Interpretation

5.1 General

With deconstruction, story network analysis and microstoria analysis as approaches in mind, the recorded interviews were listened to and the transcribed interviews were read thoroughly to become more familiar with the stories. Fragments of the stories which were relevant and which focused on the topic of investigation were identified and analyzed. The above antenarrative themes as well as the context of the study were kept in mind while the analysis was done.

Four main groups of actors were identified, namely:

1. *SU*—super user, responsible for training, handling some changes, logging errors, evaluating change requests, passing on requests to IT, sometimes directly to the external software development company (*EC*).
2. *IT*—Employee from the University's IT division.
3. *U*—User of the system, e.g. secretaries, administrative people.
4. *EC*—Employee of the external company. The development and maintenance of the IS was done by an external company.

The first three groups are part of the university environment. The external company contracted out some development work to a company from abroad (*CA*) at a specific stage of the development of the project.

For this discussion, a single interview from each of the four categories was used for analysis to ensure all categories are represented. Future work will encompass a more comprehensive analysis of all of the responses.

5.2 Analysis

Each interview was introduced by the researcher stating the background an aim of the project. The actor was then asked to elaborate on his/her experiences with the IS under study. Specific areas were highlighted, such as:

- How do/does the specific subsystem(s) that they use fit into the broader system?
- What is their involvement with this system? How do they interact with others?
- How were their experiences with and perceptions of the system, through all the phases? How were their expectations, and the comparison with the reality they encountered?
- Specific successes and/or problem situations within the IS, and the improvements that can be made.

After reading intensively through the transcribed interviews, the following three broad topics emerged:

1. The views of the actors on success or failure within systems in general.
2. Views on problems experienced and dissatisfaction with the system.
3. The role of each actor, their feelings about the systems, and the interaction with other actors.

The following sections will give some of the views of each actor on the first two above-mentioned broad topics; their perceptions on success and their views on problem situations.

Table 1 Actors' views on success/failure

Actor and views	Excerpt
SU: Put the blame on users for wanting functionalities very rapidly implemented Too little testing time may lead to failure	"Problems occurred as a result of the campuses being rushed to get new functionalities and in this way testing time was not sufficient"
IT: View is focused more towards the user	"One should adhere to the users' expectations and their workload must be reduced. The user should be empowered, service delivery must improve"
U: They want a system that must be functional and correct	"The system should work smoothly. I don't want all sorts of errors"
EC: The project manager elaborated on her views on success and failure and thereby shows her many years experience acknowledging that no IS can give 100% satisfaction to users	"Theoretically the system should be within budget, the users must be excessively happy, they must not complain about anything, but one never sees this. If a user gets 80–90% of what he wanted, then you have succeeded."
EC: viewed failure from the user's point of view and made the following statement	"The system is a failure if the user indicates that you have to stop the development because you are wasting their money"

5.2.1 Different Views on Success and Failure in Systems

When analyzing the data from each of the interviews it became clear that each actor had their own perceptions on failure and success as they have different expectations and measurements. Table 1 presents a few samples of excerpts from actors on success/failure.

5.2.2 Different Views on Problems Experienced Within the System

This section looks into the perceptions of each actor that relates to the problematic areas as they have experienced it. During the interview the actors were asked to share their experiences with the IS in question, including any problems that were encountered. Table 2 indicates some views of the actors on problems according to their own perceptions.

5.3 Discussion of Findings

The experiences of the actors were analyzed as antenarrative; no coherent flow or plot was evident. By using deconstruction analysis to look for dualities, or the other side of the story, or what has not been said, we may gain new insights into the phenomenon under investigation. Companies that develop and implement IS may learn from the views of actors involved in these systems. Although we have to be cautious in drawing general conclusions from this case, there is sufficient evidence

Table 2 Actors' views on problems

Actor and views	Excerpt
SU: They identified a few problem areas ranging from errors that occur in the system, lack of proper communication and user and management demands that cannot always be met. This *SU* group has made a plan (appointed a main user) to handle the many queries from the users, thereby protecting the time they could spend on the important issues	"The system has many errors and the system does not have stability, and one thing leads to another. We are nowhere near the 100%. As users realize there is more that systems can do, they ask for more functionalities" "The communication of errors poses a problematic situation. There are communication lines to *IT*, *EC* and different types of users. Users can be very bothersome and therefore we organized for a main user to stand between the *SU* group and the other users" "Sometimes management wants certain queries answered which are not possible to handle by the system"
IT: The view of *IT* on experienced problems was much more focused on the level of technical issues. It was acknowledged that the system was too complex. There were references to human issues such as scope and feature creep, resistance to change and training problems. Certain project management issues were also problematic	"This was a tall order for such a complex system. There were numerous interfaces; there were many errors when we went live" "We should not have tried to develop the complete system, rather used phasing in of parts" "It was problematic—the planning versus the reality at the end. You are not aware of what you don't know. There are a lot of uncertainties which make planning and estimating very difficult" "The user's expectations grow with time" "The users have resistance to change, and they panic if something is done differently" "With the new platform, the interfacing with a company from abroad was problematic in terms of communication, e.g. language and different time zones. This system is bilingual and that poses a challenge as well" "The data migration was problematic—it should have started earlier" "All the deadlines have not been met"
*U:*The user group experienced different types of frustration when using the system. These problems ranged from inadequate training to inflexibility, and from version problems, to poor system performance. A few examples of statements reflecting the views of *U* are given	"The many versions of the system are very confusing. Errors are corrected but then in the newer version other errors emerge" "The system is very slow—especially during exam times when the grades have to be input into the system" "Sometimes some information is inaccurate" "The system is not flexible, and then some lecturers use their own spreadsheets to manage the grades which are then only submitted to the system as a final grade" "Training was not adequate"

Table 2 (continued)

EC: She admitted to the complexity of the system and the need for flexibility resulting in frequent requests for changes. Improved communication and better planning were also mentioned	"Complexity of the system is high due to many functions and many subsystems"
	"New technology and software were used therefore the learning curve was high"
	"A lesson to learn was that planning must get serious attention. Work in phases"
Scope and feature creep were experienced, as users needed new functions. Certain technical matters made things worse, such as slowness of data lines and new platforms that had to be used	"It was not anticipated that data migration would be as difficult as it was in reality"
	"A lot of time was wasted waiting for the data lines (20–30% of the time) due to working from two parts of the country"
There were also political and power struggle problems as well as language barriers when a company abroad (*CA*) got the contract for a specific phase of the system development	"Politically, you have to get users to believe in the product—to buy in"
	"Language—English is a foreign language to the *CA* and they speak it with a heavy accent"
It seems that project management functions need to receive more attention	"Understanding of the academic environment needed to be communicated"
	"We had problems with estimating time allocations and budgetary issues due to many uncertainties"

to show that the different actor groups have different perceptions of the same IS in question. There are different views of the failure situations ("instability", "too many versions", "slow", "many uncertainties", "inadequate training"). Different stakeholder groups have different concerns, preferences and values. These values and concerns play a part in shaping their views of systems. Their experiences will vary based on their viewpoint and perspective. Project management recognizes the need for stakeholder engagement and the case emphasizes the differences between multiple different stakeholder groups.

The story network analysis yields the connections between the accounts of the actors. We can see that the problems experienced by *IT* and *EC* (both in the roles of project managers) are on the level of project management issues (e.g. "planning very difficult", "inadequate training", "deadlines not met", "complexity"). Regarding their views on failure and/or success *IT*, *U* and *EC* were in agreement that the users must be satisfied ("adhere to users' expectations", "system should be available", "critical functions should work"). This is significant as it goes beyond the traditional criteria used in the IS project management domain, which often focuses on internal project management targets such as time, budget and scope or functionality. As recognized by user-centered design and systems analysis, user acceptance is crucial to the success of any system and different interest groups and stakeholders recognize the need for the system to satisfy their interests and needs.

Microstoria analysis has shown that all actors' stories have to be taken into account when investigating the phenomenon. Each actor has unique problems and "lived" experiences from the specific IS. In this case it can be seen that the users may have "smaller voices" that need to be heard ("it should be fast", "always available", "too many versions", "flexibility is a problem", "main user not always available"). This again emphasizes the need for different perspectives to be acknowledged and for the recognition that different stakeholders must be brought on board to ensure buy-in that will ultimately lead to the success of systems when new IS are introduced into organizations.

6 Conclusions

Perceptions of multiple actors on IS must be acknowledge in order to get a more comprehensive view on IS success (Egorova et al. 2009; Jiang et al. 2002). Success in developing IS systems requires recognition of the different interests involved (Dalcher 2009). By analyzing the experiences of the actors in this case we see that the four groups have different perspectives which need to be taken into account to define success/failure and look beyond the traditional cost, time and feature requirements.

The following examples of guidelines can be offered to be taken into account when future IS are developed after analyzing the actors' stories as antenarrative:

- All different stakeholder groups need to be consulted—they have pluralistic views on success/failure and state different issues that they deem important.
- The number of change requests, and thereby the number of system versions, needs to be limited.
- Planning should be improved and uncertainties have to be limited or clarified.
- Communication is crucial to success; developers should work towards improving communication lines between all actors.
- Training should be planned and executed to ensure that the users will be empowered to participate and use the system.

The value of antenarrative approaches is shown as identifying the problem areas within this IS that is normally not easily obtained through traditional approaches such as post project evaluations. We now look into the small stories and the unsaid issues emerge from studying the accounts given by the actors. However, we cannot generalize, but we might use insights gained from this study to apply to other software systems as well. Future work includes the analysis of all of the collected data, and application of a few narrative analysis approaches thereby comparing each method's value, depth and insights in the results obtained.

The lessons learned from the application of the methods described above can support the development of new information systems and can play a part in beginning to address the concerns relating to the high rate of failure in information systems. The contribution of this study is in the area of development practice, as well as in the area of narrative theory and practice—the discipline that this study has borrowed from. The insights from the proposed future work may further help to refine and define our view of success and failure in Information Systems.

Acknowledgements This paper is based upon work that was financially supported by the NRF. The opinions expressed in this paper are those of the authors.

References

Anon (2007) What determines a project success or failure? www.lessons-from-history.com. Accessed 31 Oct 2007

Boje DM (2001) Narrative methods for organizational and communication research. Sage, London

Clandinin D, Connelly F (2000) Narrative inquiry: experience and story in qualitative research. Jossey-Bass, San Francisco

Cortazzi M (1993) Narrative analysis. Falmer Press, London

Czarniawska B (1998) A narrative approach to organization studies. Sage, Thousand Oaks

Dalcher D (2004) Stories and histories: case study research (and beyond) in information systems failures. In: Whitman ME (ed) The handbook of information system research. Idea Group Inc. (IGI), London, pp 305–322

Dalcher D (2009) Software project success: moving beyond failure. UPGRADE 10(5):42–50 (Oct 2009)

Drevin L, Dalcher D (2011) Antenarrative and narrative: the experiences of actors involved in the development and use of information systems. In: Boje D (ed) Storytelling and the future of organizations: an antenarrative handbook. Routledge, New York

Egorova E, Torchiano M, Morisio M, Wohlin C, Aurum A, Svensson RB (2009) Stakeholders' perception of success: an empirical investigation. In: Proceedings of 35th EuroMicro conference on software engineering and advanced applications. Patras, Greece, pp 210–216

Fortune J, Peters G (2005) Information systems: achieving success by avoiding failure. Wiley, Chichester

Harmer B, Pauleen D (2007) The work anytime anywhere mobility blues. In: 4th QAULIT conference. Wellington, New Zealand

Jiang JJ, Klein G, Discenza R (2002) Perception differences of software success: provider and user views of system metrics. J Syst Softw 63(1):17–27 (July 2002)

KPMG (2005) Global IT project management survey. http://www.pmichapters-australia.org.au/canberra/documents/irmprm-global-it-pm-survey2005.pdf. Accessed 31 Oct 2007

Lieblich A, Tuval-Mashiach R, Zilber T (1998) Narrative research. Sage, Thousand Oaks

Lyhne L (2007) eNatis crash may be due to missing safety net. Sunday Times. http://dariel.co.za/news6.html. Accessed 19 Oct 2007

Mishler EG (1995) Models of narrative analysis: a typology. J Narrat Life Hist 5(2):87–123

Riessman CK (1993) Narrative analysis. Sage, Newbury Park

Standish (1999) A recipe for success. http://www.standishgroup.com/sample_research/PDFpages/chaos1999.pdf. Accessed 8 Mar 2004

Standish (2001) Extreme chaos. http://www.standishgroup.com/sample_research/PDFpages/extreme_chaos.pdf. Accessed 8 Mar 2004

Standish (2003) Latest Standish Group Chaos report shows project success rates have improved by 50%. http://www.standishgroup.com/pres/article.php?id=2. Accessed 23 Feb 2004

Standish (2004) 2004 third quarter research report. http://www.standishgroup.com/sample_research/PDFpages/q3-spotlight.pdf. Accessed 25 Apr 2006

Telegraph (2007) £20bn NHS computer system doomed to fail. http://www.telegraph.co.uk dated 13-2-2007. Accessed 19 Sept 2007

Towards a Unified Service Definition

George Feuerlicht and Lukas Burkon

Abstract Services play an increasingly important role in most industry sectors, including IT. Recent emergence of Cloud Computing and the general acceptance of different forms of outsourcing as a mechanism for increasing effectiveness of organizations necessitate a revision of basic SOA concepts to take account of extensive use of externally provided services. A key pre-requisite for this revised SOA architectural framework is the unification of service definitions so that different types of services can be treated in a consistent manner. The unified service definition needs to contain attributes of internally and externally provided services as well as attributes of business and IT services and be customizable to allow its application in different usage scenarios. In this paper we discuss the concept of a service as interpreted in both business and computing literature and propose a concise service definition based on a set of attributes that can be specialized for a particular service delivery scenario. We illustrate using a simple example how such service definition can be used to construct instances of services.

1 Introduction

Services account for a growing percentage of GDP (Gross Domestic Product) of advanced economies and are creating new opportunities for innovative business models. The Information Technology (IT) industry plays an important role in this trend towards service economy and is itself being transformed into a service-based industry with many organizations adopting Service-Oriented Architecture (SOA) [1]. Most services today involve some level of IT support and can be regarded as aggregations

G. Feuerlicht (✉)
Department of Information Technology, University of Economics, W. Churchill Sq. 4, Prague, Czech Republic
e-mail: jirif@vse.cz

Faculty of Engineering and Information Technology, University of Technology, P.O. Box 123, Broadway, NSW 2007, Sydney, Australia
e-mail: george.feuerlicht@uts.edu.au

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_18,

of business and IT services. For example, a courier service has a non-IT component—the actual delivery of a parcel by a courier to a specified address, and is supported by a parcel tracking application accessible over the Web that represents the IT component of this service. It is this interplay of business and IT services that often produces innovative business processes that can give an organization competitive advantage. Business process outsourcing is an important mechanism for improving the overall effectiveness of the organization and an imaginative outsourcing strategy can lead to a rapid expansion of the business. However, effective outsourcing decisions require good understanding of the overall service model of the organization, in particular the interaction between various services that constitute this model, and a careful analysis of the benefits and drawbacks of externally sourced services. While most business services today are associated with some type of a computer application, it is useful to model services that do not involve IT as an integral part of the overall service framework. In practice, the difference between IT and business services is not always easily discernable as in many cases a business component of a service is closely related to the supporting software that implements the service (e.g. in electronic banking), making the classification of services as business or IT difficult.

The increasing popularity of various types of subscription models such as SaaS (Software as a Service), PaaS (Platform as a Services) and IaaS (Infrastructure as a Service), generally known under the term Cloud Computing is further accentuating the trend towards service-based enterprise computing [2]. However, the adoption of these service models introduces its own set of problems as organizations need to be able to integrate and manage services potentially sourced from a number of different providers and to incorporate these services into their existing Enterprise Architecture (EA). The emerging enterprise computing scenario involves organizations interacting with multiple external providers of various types of ICT services, as well as with partner organizations within the context of supply chains and other types of collaborative scenarios that include social networking platforms, Wikis, Blogs, mashups, and Web APIs, changing the character of enterprise applications (Fig. 1).

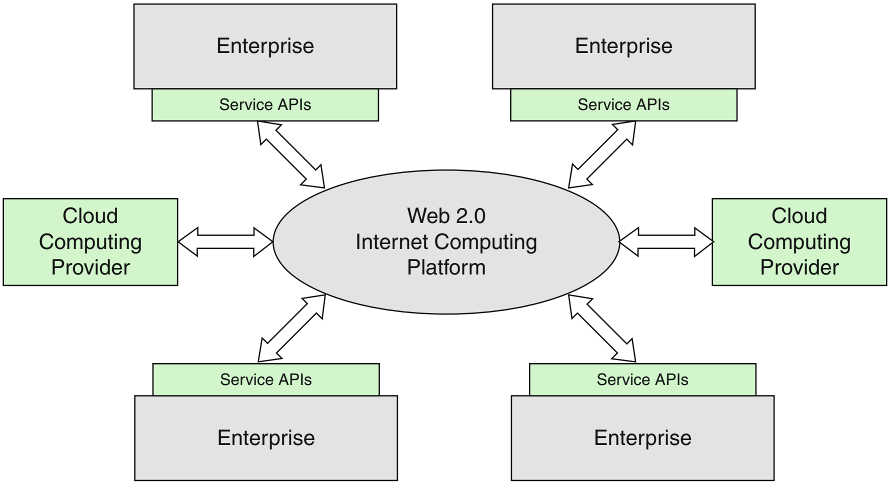


Fig. 1 Emerging enterprise computing environment

As the adoption of Cloud Computing accelerates, it is likely that most enterprise computing needs will be addressed by external providers of services, and internally organizations will be mainly concerned with service integration and with managing the interfaces with services providers. This allows organizations to focus on their core business activities, freeing end-user organizations from the burden of having to develop and maintain large-scale IT systems. Some progressive organizations are already aggressively pursuing outsourcing strategy based on SOA. For example, the low-cost Australian airline Jetstar (www.jetstar.com) is in the process of building a franchise network across Asia, heavily relying on outsourcing to facilitate rapid expansion [3]. Most of Jetstar's business functions are outsourced including renting of aircrafts, in-flight catering, and overseas ground handling. The payroll system is a SaaS application, and IT infrastructure and support are fully outsourced to external providers. Virtualization plays a major role within Jetstar and enables fast deployment of check-in applications on mobile check-in units in regional airports using a Citrix-based thin-client solution. All software applications including Jetstar Virtual "Office", Email, Staff Rostering and Schedules, Staff Rewards, and Employee Kiosk are implemented as virtualized application services deployable anywhere where Jetstar operates with minimum expense and delay. The entire IT operation relies on small team of IT professionals who are primarily responsible for managing the relationship with individual service providers. Managing such highly complex and rapidly evolving business environments represent a challenge as the development an effective outsourcing strategy requires an enterprise architecture with consistent representation of services irrespective of the type of service and its origin, so that a large number of variants with different critical success factors can be evaluated [4].

Traditionally, EA research focuses on alignment of business and ICT strategy [5, 6]. More recent methodologies adopt the SOA approach and focus on the management of life-cycle of ICT services [7, 8]. The adoption of SOA concepts and principles creates an opportunity to incorporate externally provided services into the architectural framework and to extend the scope of the methodology to unify the management of internal and externally sourced ICT services. Although the concept of SOA has been evolving over time, it still remains focused on intra-enterprise computing and does not fully address the challenges associated with the externally provided IT services. The basic underlying SOA principles and assumption need to be re-evaluated to encompass externally provided software and infrastructure services, and to enable organizations to make informed decisions about various types of outsourcing models. Effective management of services requires an architecture that treats different types of services in a unified manner so that a consistent model of the entire organization including externally sourced services can be developed. A key pre-requisite for this revised SOA architectural framework is the unification of service definitions so that different types of services can be treated in a consistent manner. Most current methodologies e.g. IBM SOMA (Service-Oriented Modelling and Architecture) [9] focus on modelling internal, on-premise services and do not satisfactorily address situations that involve externally provided Cloud Computing services. Other methodologies (e.g. ITIL) [10] focus exclusively on services that involve IT components (i.e. services supported with software applications) and do not provide a suitable mechanism to represent purely business services in the overall service model.

In this paper we firstly discuss the concept of a service as interpreted in both business and computing literature and review existing work on service description (Sect. 2). We then illustrate the need for a unified service description using a simple example (Sect. 3), and describe our proposal based on a set of attributes that can be specialized for a particular service delivery scenario, illustrating how such service description can be used to construct service instances (Sect. 4). The final section (Sect. 5) summarizes the contribution of this paper and outlines further work.

2 Related Work

Recent trend towards service-orientation in business in general, and in IT in particular has resulted in an extensive research effort to incorporate service-oriented concepts into existing frameworks and methodologies. For example, the latest version of the ITIL v3 (Information Technology Infrastructure Library) [10] includes best practices for design, development, and implementation of service management. Similarly, ITSM (IT Service Management) methodology addresses management of IT services by applying SOA concepts with the aim to create agile IT environment based on Web Services and related technologies [11]. Academic efforts include the SPSPR model designed to integrate IT and business processes using a layered architecture [12]. Services Science is another area of research effort concerned with services from a more traditional business perspective attempting to describe the basic patterns for the development and deployment of services. Industry participation in Services Science research includes IBM, HP and BT in collaboration with several UK universities in the context of the UK Network of Researchers in Services Sciences [13, 14].

Numerous definition of business services are used in the literature. While some attempts are made at treating services as both business and IT concepts, in general, most research efforts are conducted separately, either in business or in computing literature. For example, the ITIL specification [7] differentiates between business and IT services, defining business services as services implemented or supported by IT services (i.e. ITIL does not consider business services without IT support). Business service is defined as: "A means of delivering value to customers by facilitating outcomes customers want to achieve without the ownership of specific costs and risks." On the other hand, ITIL IT service is defined as: "A service provided to one or more customers by an IT service provider. An IT service is based on the use of IT and supports the customer's business processes. An IT service is made up from a combination of people, processes and technology and should be defined in a Service Level Agreement." Other authors define IT (ICT) services as: "ICT services are activities and/or information delivered by the ICT service provider to a service receiver (customer).", emphasizing the relationship between the recipient and provider of ICT services [12].

Another comprehensive source of service specifications is the area of Web Services standardization and related literature. W3C defines Web Service as: "A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards." [15]. Although the definition of Web Services is more technical than previously considered service definitions and relies on existing language and protocol standards (i.e. HTTP and XML), Web Services standardization, in particular the WSDL (Web Services Description Language) standard [16], and higher-level standards such as BPEL, WS-Policy, WS-Security, and WS-Resource provide a useful guide for the formulation of the concept of service and for detailed service description [17].

The COSMO framework [18] deals with service abstraction from a design point of view and offers several options for service design at different levels of abstraction. Service interactions are modeled at three levels of abstraction: (1) simple interaction (service provides a direct value to the recipient in the simple form) (2) choreography (service is a rich system of interactions between the recipient and provider), and (3) orchestration (service is implemented as a composition of other services, and the service provider acts as the interlinking coordinator). Similar to BPEL, recursive composition of services produces higher level services, resulting in composite services and networks of interconnected services. In a survey of the general nature of services the authors highlight the importance of non-functional service properties and analyze the lifecycle of service from the perspective of the service provider, the service requestor and service broker [19]. Non-functional properties include temporal and spatial availability, charging styles, settlement models, settlement contracts, payment, service quality, security, trust and ownership. The paper contains a detailed discussion of how non-functional service properties are used for operations during the various phases of service lifecycle (i.e. discovery, substitution, composition, etc.).

While there is extensive research effort in both business and IT domains, there are relatively few attempts to combine business and IT services into a single consistent architectural framework. From the IT perspective, services are defined in the context of SOA, and while IT services are typically encapsulated into higher level business functions the primary focus remains on IT and no support is provided for business functions that are not supported by underlying IT applications. The business perspective, on the other hand, tends to deal purely with the business services focusing on marketing, customer relationship management, and other business functions without providing a mechanism for the description of IT services to a sufficient level of technical detail. Given this separation into business and IT services it is difficult to provide a coherent description of the entire organization using a consistent service model capturing the interplay between business services and the underlying technological services that support the various business processes.

Unification of the concept and specification of services is a starting point for resolving this dichotomy. Using a unified service definition makes the analysis, design and composition of services easier to perform as all services are specified using a single set of characteristics. Service catalogues and dictionaries become more comprehensible and easier to search, and existing services can be incorporated into new service compositions without the need to resolve inconsistencies between different types of service specifications.

3 Motivating Example

Consider, for example a simplified flower shop business scenario illustrated in Fig. 2. Flowers Only (Stage 1) is a small flower shop specializing in flower arrangements for birthdays and other occasions. Customers order flowers over the phone and Flowers Only organizes the packaging and delivery of selected flower arrangement. Flowers Only is responsible for all business functions, including growing the flowers, accounting and payroll. These business functions are illustrated in Fig. 2 as internal services.

In Stage 2 of its evolution Flowers Only decides to expand the business and implements an online eShop and at the same time outsources the payroll function to a SaaS provider and the packaging and delivery function to a courier company. This new situation is illustrated in Fig. 3, showing Payroll, Packaging & Delivery, and eShop as externally sourced services. Remaining business functions are implemented as internal services.

To allow further expansion of the business (Stage 3), Flowers Only decides to offer additional products including bottles of selected wines that can be packaged together with flowers or delivered as separate items. To enable expansion and management of its customer base, Flowers Only signs up for a SaaS CRM (Customer Relationship

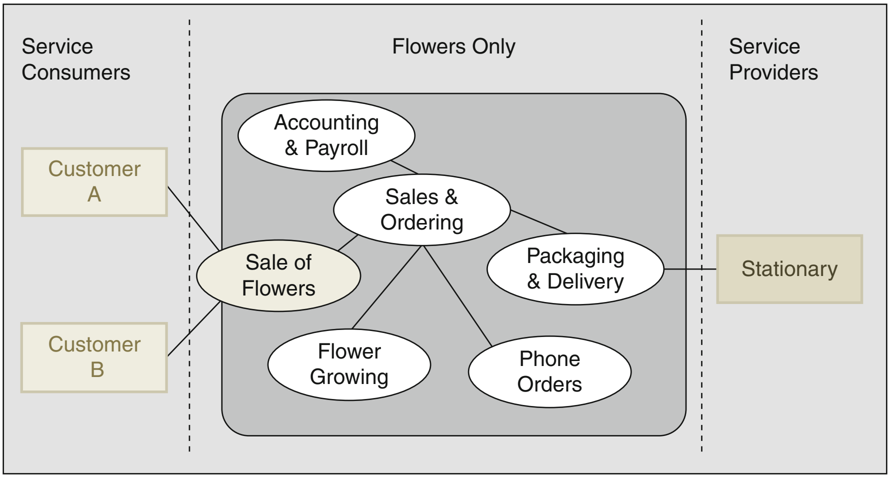


Fig. 2 Flowers only—Stage 1

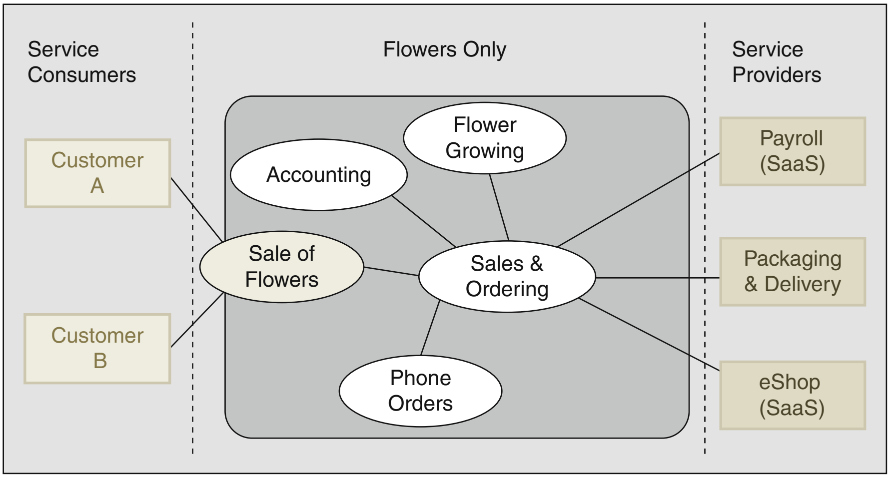


Fig. 3 Flowers only—Stage 2

Management) application. This new situation is illustrated in Fig. 4, showing that most services are now sourced externally, allowing Flowers Only to concentrate on its core business and future expansion into additional services for its customers. This rather simple example illustrates a typical situation in many organizations today that need to respond to a rapidly changing business environment and exploit new business opportunities as they arise. In practice, the scale of the problem is much greater than is indicated by this simple scenario as large organizations need to manage more complex service environments and at the same time attempt to maximize return on investments by making informed outsourcing decisions.

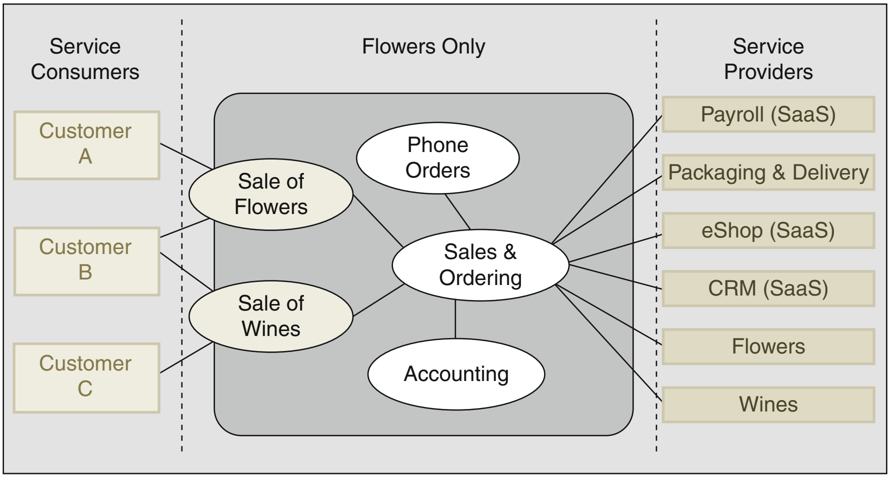


Fig. 4 Flowers only—Stage 3

4 Unifying Service Description

In this section we consider service description in more detail drawing on various approaches available in the literature [12, 18–20]. Service interface specifies the content and purpose of the service and conditions associated with its delivery. Precise definition of the service interface facilitates communication between service provider and service consumer and constitutes a contract between the two parties. In the case of business services, service interface is usually specified in a semi-structured form using a SLA (Service Level Agreement) or OLA (Operating Level Agreement). More technical service specifications such as WSDL use interface specification languages with precisely defined structure and content, and based on agreed semantics of operations. Service interface specification consists of a set of attributes that represent the various properties of the service; these attributes can be either atomic or can be structured so that the resulting specification constitutes a taxonomic hierarchy of attributes. The following set of top level attributes (Table 1) has been adapted from literature sources [12, 18–20].

Service interface attributes shown in Table 1 include the service identifier and service description (functional parameters describing the service operations), service provider and consumer, a range of non-functional attributes (cost, quantity and quality of service) and runtime attributes (provisioning and protocol).

Table 1 Service interface attributes

Service attribute	Attribute description
Identifier	Attribute, or a set of attributes that uniquely identifies the service
Description	Full description of the service that includes objective of the service, its functional parameters, etc.
Provider	Service provider organization in the case of a business service, or the address of a resource, e.g. WSDL port for a Web Service
Consumer	Service consumer organization in the case of a business service, or a software client, in the case of Web Services or other types of software services
Cost	Cost of the service, discount policy, etc.—can take the form of a subscription payment or per transaction payment. In some cases the service is free of charge
Quantity	Volume of the service—can be expressed as the number of users of the service per unit of time, volume of data stored, etc.
Quality	Quality of Service (QoS) parameters include service availability, response time, service guarantees, but also other metrics such as customer satisfaction, etc.
Provisioning	Service authentication, enrollment, metering, billing, expiration time, etc. For business services, provisioning also includes legislative options and constraints that specify the conditions under which the service is made available, e.g. security restrictions, etc.
Protocol	Runtime dialog that controls the service behavior, i.e. service process model that defines responsibilities of the service provider and service consumer

4.1 Service Interface Specialization and Service Instances

The above description of service interface attributes provides a basic template for the specification of the service. In most practical situations this description needs to be further refined by specifying additional sub-structure for at least some of the attributes. For example, the Cost attribute can be substituted by a more specific pricing model for the service that defines discounts for categories of service consumers and includes default values that apply to specific situations. This attribute refinement can be applied recursively resulting in a substructure of attributes tailored to a particular service context.

4.1.1 Service Instances

Service instance is an occurrence of a specific service type, i.e. the instantiation of a service type for a particular pair of service provider and service consumer, given a specific context. For business services this may involve a long-term contractual arrangement specified by a detailed SLA, in the case of Web Services this relationship tends to be more transient and may involve consuming a service without creating ongoing relationship between the service provider and consumer (e.g. stock value lookup using a Web Service). Service instances can use default values specified for the service type or override these defaults by attribute values supplied for a particular service occurrence. These instance values must, however, conform to the policies and constraints of the service specification (e.g. the service cost may be fixed, and no variations allowed). Service instantiation may be preceded by a negotiations phase with the service consumer actively seeking to vary the default conditions (i.e. attribute values) of the standard contract. While negotiating conditions under which a service is provided is more typical for business services, it is possible to use the same principle for services implemented as software. In some cases negotiations may result in a new service sub-type with additional service interface attributes.

Returning to our Flowers Only example introduced in Sect. 2, we illustrate how the service interface can be defined for a specific sale of flowers scenario, starting with a high-level (generic) service interface specification (Table 2). This generic service interface can now be specialized to include various default flower arrangements:

1. Red Rose Bouquet (Cost = $ 79, Quantity = six roses)
2. Gorgeous Lily Bouquet (Cost = $ 90, Quantity = five stems of wrapped lilies)
3. Mixed Flowers Basket (Cost = $ 99, Quantity = basket of six kinds of flowers)

A service instance for the Gorgeous Lily Bouquet flower arrangement is illustrated in the Table 3.

The above service instance illustrated in Table 3 corresponds to a specific service identified by Order Number 000001 and is of specialized type "Gorgeous Lily Bouquet".

Table 2 Service interface specification for flower only

Service interface for sale of flowers	
Identifier	Sale of flowers
Description	Sale of fresh flowers to customers
Provider	Roses only
Consumer	Individuals or companies
Cost	The service cost depends on the type and number of flowers ordered
Quantity	Number of flowers
Quality	The service quality can be measured by the freshness of the flowers and speed of delivery
Provisioning	The flowers are delivered to most business addresses by 5:00 pm and most residential addresses by 7:00 pm, in most major Australian metropolitan areas for orders received by 8:00 am
Protocol	Orders can be placed online via the eShop or by phone specifying the type and number of flowers and the delivery address. Following a payment, email acknowledgement is sent and the flowers are delivered to the customer address

Table 3 Flower only service instance

Instance of sale of flowers service (Gorgeous Lily Bouquet)	
Identifier	Sale of flowers (order number 000001)
Description	Sale of fresh flowers to customers—Gorgeous Lily Bouquet
Provider	Roses only
Consumer	Lukas
Cost	$ 90
Quantity	Five stems of wrapped lilies
Quality	Fresh flowers delivered by 7:00 pm
Provisioning	The flowers by 7:00 pm, in most major Australian metropolitan areas
Protocol	The order was placed online via the eShop specifying the type and number of flowers and the delivery address. After receiving a payment email acknowledgement was sent and the flowers delivered to the customer address

5 Conclusions and Further Work

Consistent and extendable definition of service interfaces is the first step towards developing architectural support for complex service-based environments that are becoming widespread today. The unification of description of the various service concepts is essential for modeling and integration of services of different type. Using a consistent approach to describe different types of services will ensure mutual compatibility of service interfaces and provide a mechanism for service composition. Service providers can use this framework to produce variants of service specification by extending the attribute set of the service descriptors, facilitating more effective management of services.

In this paper we have argued for the need to model services using a consistent framework irrespective of the type of service, i.e. if the service is a business service or an IT service, or a combination of different service types. Importantly, the modeling framework should accommodate services implemented internally within the organization and services sourced from external service providers. We have proposed a simple set of attributes that provide a starting point for service description that can be applied to services of different types and specialized as required by a particular business scenario. We have illustrated using a simplified example how such service definition can be used to construct service instances.

This paper presents some initial ideas on unifying service definitions for different types of services and at the same time supporting extendibility via specialization of the specification. Further work is required to refine the service framework by incorporating additional attributes to account for specific requirements of IT services utilizing sources such as Web Services standard specifications (e.g. WS-Agreement) and existing SOA modeling methodologies. Moreover, applying the approach described in this paper to a more comprehensive example that involves a larger number of services of different type will identify additional issues that need to be addressed to provide a consistent framework for modeling, composition and integration of services.

Acknowledgements This research was supported by IGA (Internal Grant Agency) grant No. F4/1/2010, GAČR (Grant Agency, Czech Republic) grant No. P403-10-0092, and the Research Centre for Human Centered Technology Design at the University of Technology, Sydney.

References

1. Sholler D (2008) 2008 SOA user survey: adoption trends and characteristics. Gartner, 2008:32 (26 Sept 2008)
2. Armbrust M et al (2010) A view of cloud computing. Commun ACM 53(4):50–58, ISSN 0001-0782
3. Tame S (2008) Implementing IT and SOA into the Jetstar business. In: SOA 2008 IQPC conference, Sydney, Australia
4. Vorisek J, Feuerlicht G (2006) Trends in the delivery and utilization of enterprise ICT. In: Proceedings of 17th annual IRMA international conference, Washington DC, USA, 21–24 May 2006
5. Peppard J, Ward J (2004) Beyond strategic information systems: towards an IS capability. J Strategic Inf Syst 13(2):167–194
6. Langenberg K, Wegmann A (2004) Enterprise architecture: what aspects is current research targeting. In: EFPL Technical report IC/2004/77, Ecole Polytechnique Federale de Laussane
7. OGC (2007) The introduction to the ITIL service lifecycle: Office of Government Commerce (official introduction). The Stationery Office, London, p 238. ISBN 978-0-11-331061-6
8. Brand K, Boonen H (2008) IT governance based on CobiT 4.1—a management guide. Centraal Boekhuis, Culemborg. ISBN 9087531168
9. Arsanjani A et al (2008) SOMA: a method for developing service-oriented solutions. IBM Syst J 47(3):377–396
10. Van Bon J, de Jong A (2007) IT service management: an introduction based on ISO 20000 and ITIL V3. Van Haren Publishing, Norwich. ISBN 9789087530518
11. O'Neill P, Krauss D (2006) Topic overview: IT service management. Forrester Research, Cambridge
12. Voříšek, J, (2008) Principles and models for the management of enterprise ICT. Oeconomica, Praha. ISBN 978-80-245-1440-6

13. Hefley B, Murphy W (2008) Service science, management and engineering: education for the 21st century. Springer, Heidelberg. ISBN 978-0-387-76577-8
14. Macaulay L, Zhao L (2008) SSME at Manchester: bringing people, business and technology together. In: IBM SSME summit: education for the 21st century, New York
15. Haas H, Brown A (2004) Web services glossary, W3C working group note. World Wide Web Consortium (W3C), MIT/CSAIL, Cambridge
16. Christensen E et al (2001) Web services description language (WSDL) 1.1. http://www.w3.org/TR/wsdl. Accessed 10 Mar 2010
17. Web Services Activities (2010) http://www.w3.org/2002/ws/. Accessed 10 Mar 2010
18. Quartel DAC et al (2007) COSMO: a conceptual framework for service modelling and refinement. Inf Syst Frontiers 9(2):225–244
19. O'Sullivan J, Edmond D, ter Hofstede AHM (2003) Service description: a survey of the general nature of services. Technical report FIT-TR-2003-02, Queensland University of Technology, Brisbane
20. Alter S (2008) Service system fundamentals: work system, value chain, and life cycle. IBM Syst J 47(1):71–86. ISSN 00188670

A Process-Oriented Approach to Support Dynamic Evolution in Distributed Applications

Kam Hay Fung and Graham Low

Abstract Dynamic evolution is beneficial and important to certain classes of distributed applications where shutdown is not always viable requiring modifications to such applications while they are executing. This paper describes Continuum, a process-oriented approach to address analysis and design aspects of dynamic evolution in software development, leveraging the flexibility of composition-based applications—like component-based and service-oriented ones—to facilitate dynamic changes. The approach begins with analyzing and extending the application lifecycle of an application to accommodate dynamic changes (i.e. while it is still executing). Then, the design of runtime modifications is carried out to progressively realize the changes, in terms of what to perform (i.e. transformations), who performs the transformations (i.e. transformation agents), and how the transformations are performed (i.e. transformation steps). A hypothetical electronic product catalogue application is used to illustrate Continuum.

1 Introduction

Dynamic evolution is a phenomenon whereby changes are applied to an application during runtime without the need for shutdown and restart [9]. The ability to evolve with minimal interruption appeals to certain types of applications in which shutting them down for changes is not always a desirable option (e.g. loss of US$ 10–15k per minute [11] due to downtime).

Composition-based software development refers to the paradigm of "weaving" existing or custom-developed systems, commercial-off-the-shelf software components and applications, leased or rented, into composite systems-of-systems [6]. Composition-based applications—component-based, service-oriented, etc.—are well suited to dynamic evolution because of the flexibility of their structure

K. H. Fung (✉)
School of Information Systems, Technology and Management,
The University of New South Wales, Kensington, NSW, Australia
e-mail: kamhayfung@ieee.org

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_19,

which is due to their fabrication from loosely coupled *parts* (of functionality) and their *bindings* (for connecting parts). Operations such as addition, replacement and removal of parts are performed to accommodate ongoing changes to these applications. Moreover, the notion of distribution in which application parts are scattered and interact over a network further facilitates these operations by connecting (new) parts to and disconnecting (existing) parts from an application.

Studies recommend that evolution in applications should be considered in a development methodology [2] which involves a process to follow, work products to create and use, and consideration of people and tools [10]. While many studies investigate techniques and tools that could be useful in dynamic evolution, the methodological aspect has been largely ignored. Fung and Low [8] evaluated twelve methodologies supporting composition-based distributed application development. The results indicate immaturity in these methodologies for dynamic evolution. In related work, the *staged lifecycle* model [2] expresses an application's lifecycle as going through several stages from initial development to close-down. However, it makes no provision for analyzing and evolving a running application to accommodate dynamic changes. Oriezy et al. [14] use two process lifecycles to deal with dynamic self-adaptation in software. Their approach emphasizes particular architectural styles to embrace changes but lacks details on implementing changes.

This research contributes to previous efforts on dynamic evolution by taking a first-step towards a methodological approach to support dynamic evolution during the analysis and design of software development. In particular, Continuum, proposed herein, is a methodology extension comprising a set of interrelated features to specifically address analysis and design aspects of dynamic evolution. Due to space limitations, this paper concentrates on the process aspects of Continuum (at a high-level). Full descriptions of Continuum are not included in this paper.

The rest of this paper is organized as follows. Section 2 describes a hypothetical application to guide the discussion of Continuum in this paper. Section 3 introduces key concepts underpinning Continuum to reason about dynamic evolution. Section 4 presents Continuum processes and how they fit together in a development lifecycle. Section 5 discusses our preliminary evaluation on Continuum. Section 6 concludes with future research work planned for dynamic evolution.

2 Electronic Product Catalogue Platform

To illustrate the process features in Continuum, we refer to a hypothetical application called Electronic Product Catalogue Platform (EPCP) throughout the paper. EPCP serves electronic product catalogues (EPC) on items of interest (e.g. merchandise) via the web, comprising (shown in Fig. 1)

- *Web User Interface* (*WebUI*): for end users to enter search criteria and view search results.

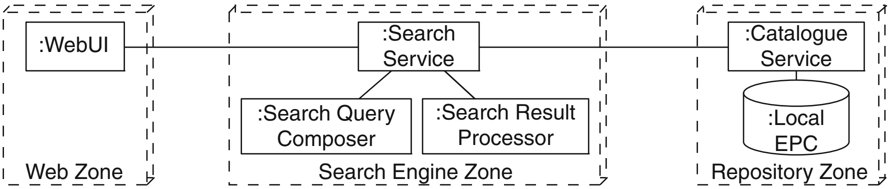


Fig. 1 EPCP: current structure

- *Search Service*: coordinates search invocations to Catalogue Service. It uses *Search Query Composer* to build search queries for entered search criteria, and *Search Result Processor* to rank and iterate through search results.
- *Catalogue Service*: provides access to EPC data.
- *Local EPC*: a database providing up-to-date catalogue information on items of interest, offered by an organization in local currency.

Suppose that after the initial successful launch of EPCP, its stakeholders have decided to have EPCP *integrated with two external partner EPC systems* to provide additional catalogues for an increased variety of merchandise. Furthermore, EPCP will also support the *internationalization* (a.k.a. "i18n") capability [1], meaning EPCP will handle catalogue items described in foreign languages and priced in different currencies. Since EPCP is running around the clock, it is desirable to progressively roll out these changes while reducing interruption to EPCP.

3 Key Dynamic Evolution Concepts

Continuum divides the notion of dynamic evolution into three concerns: application lifecycle, transitional period and transformation. Each concern symbolizes a particular aspect of dynamic evolution on a different time scale to handle complexity. On a coarse-grained time scale, the *Application Lifecycle* concern characterizes the progression of an application over its generations of releases during its operating lifespan. A *Generation* denotes the application is operating a particular version of its code, such as Fig. 1 which shows the current generation of EPCP. The notion of *Transitional Period* represents the period in the application lifecycle during which the application advances from one generation to the next via high-level runtime modifications called *Transformations*. In a distributed environment, transformations are collaboratively performed by *Transformation Agents* over the network. On a fine-grained time scale, the Transformation concern prescribes low-level atomic runtime modification steps to the application's elements, such as initializing a new component before use.

In addition to the aforementioned key concepts, Continuum prescribes *Transformable Items* and *Zones* as abstractions for a composition-based distributed application. A Transformable Item denotes a logical entity in a distributed application at runtime. It epitomizes a part, a binding, or a composite thereof, and can be transformed into some other form. A Zone [7] defines a disjoint partition (e.g. a logical node in a network) for hosting Transformable Items. In the EPCP example, its elements—q.e. transformable items—are scattered over three zones: Web, Search Engine, and Repository (Fig. 1).

4 Continuum Processes

In light of the approach to divide dynamic evolution into three concerns, viz. application lifecycle, transitional period and transformation, Continuum prescribes four dynamic evolution specific processes for the analysis and design of changes to composition-based distributed applications.

- *Application Lifecycle Analysis*, to address the application lifecycle concern during analysis;
- *Transformation Identification* and *Transformation Agent Design*, to tackle identification, sequencing and assignment of transformations during a transitional period; and
- *Transformation Design*, to handle the detailed design of each transformation.

Continuum processes are expected to be used in the context of an iterative and incremental development approach driving ongoing changes to an application as they arise. Conventional development aspects such as testing are expected to be handled by features of existing methodologies. To utilize Continuum processes, a process lifecycle model is constructed to include these processes. Alternatively, they are incorporated into an existing lifecycle model such as via method engineering [4]. Figure 2 is one such lifecycle model with two concurrent streams of development activities, one for conventional application development (lower half) and the other devoted to dynamic evolution (upper half). It begins with analysis processes on the left, in which a conventional Requirement Analysis process elicits, prioritizes, scopes

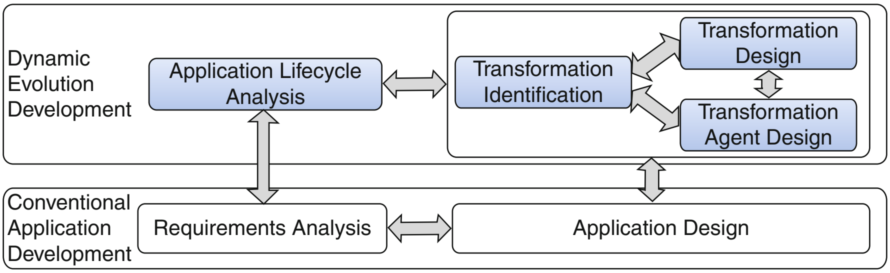


Fig. 2 Example analysis and design project lifecycle with Continuum processes

and approves requirements and change requests in an endeavor. Continuum's Application Lifecycle Analysis process runs alongside Requirement Analysis to take approved requirements and change requests from the latter as inputs, and to plan for the application lifecycle and successive generations necessary to fulfill those requirements and change requests. This process also converts those requirements and change requests into change specifications, called *change cases* [5]. In Continuum, change cases are confined to articulating dynamic changes to elements of an application, its structure and its zones (e.g. "Add i18n support to WebUI" for EPCP).

Requirements and change requests are fed to a conventional Application Design process which focuses on leveling the static design of the application to meet the new requirements and change requests with little regard for dynamic evolution. Likewise, change cases are fed to Continuum's Transformation Identification process which identifies the transformations required to promote the running application to the runtime structure based on the new design. Next, one can continue with Continuum's Transformation Design process and then Continuum's Transformation Agent Design process. In this case, the detailed design of all transformations is worked out before transformation agents are identified and assigned to these transformations. Alternatively, one continues with Transformation Agent Design followed by Transformation Design, in which case transformation agents are determined before the detailed design of individual transformations. In the following sections, Continuum processes are described in detail.

4.1 Application Lifecycle Analysis

This process aims to define the roadmap for an application as successive stages in its lifecycle that will accommodate changes elicited from requirements analysis. It does so by extending its lifecycle with new generations, with each transitional period between two successive generations labeled with associated changes to be accomplished during the period.

To illustrate this process, consider the two enhancement requirements stated earlier in Sect. 2, being the i18n capability and integration with external EPC systems for EPCP. The first step in this process is to translate these requirements into change cases. According to the architecture (Fig. 1), suppose that Catalogue Service is to handle catalogues in different currencies and languages and to plug-in two external EPC systems. In the Web Zone, WebUI will be extended for end users to enter search criteria for and display catalogue items in different currencies and languages. The resulting change cases are specified in Table 1.

Next, the application lifecycle of EPCP is extended to incorporate these change cases. To determine which change cases should be accomplished first, an analysis of the nature and purpose of the change cases is given. Since the two external EPC systems will hold catalogue items in a foreign language, the EPCP application must support i18n before it is integrated with the EPCs (i.e. both CC31 and CC21 realized first). On the other hand, before Catalogue Service serves catalogue items

Table 1 EPCP: key change cases

Change case ID	Purpose	Description
CC11	Add i18n support to WebUI	WebUI extended to display a variety of products in different units of currency and languages
CC21	Modify Search Service to integrate with new Catalogue Service	Search Service modified to interact with the new Catalogue Service instead of the old one
CC31	Add i18n support to Catalogue Service	Catalogue Service extended to support catalogue data in a variety of currencies and languages
CC32	Modify Catalogue Service to integrate with two external EPC systems	Catalogue Service modified to integrate with two external EPC systems that are operated by business partners

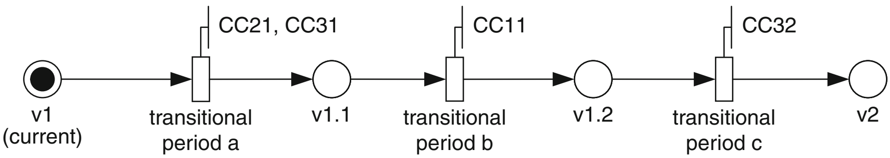


Fig. 3 EPCP: application lifecycle diagram

from external EPCs to WebUI, WebUI must be able to display them in foreign languages/currencies. This analysis suggests a sequential order in which the changes cases should be realized: from CC21/CC31 to CC11 and then followed by CC32. Thus, the application lifecycle is extended taking into account this constraint. The resulting configuration is shown in Fig. 3 in Petri Net notations [12]. The lifecycle designates three transitional periods—noted as a, b and c—to roll out the annotated change cases. Generation v2 on the right represents the final outcome of dynamic evolution, involving the passing of two intermediate generations, v1.1 and v1.2, one enhanced with the i18n capability and the other enhanced with foreign currency and language support.

4.2 Transformation Identification

During design, this process aims at identifying a set of transformations to be executed in a transitional period to advance an application from an as-is generation to a to-be generation that will have accommodated the proposed changes for the as-is generation. In this respect, the first step in this process is to identify the as-is and to-be generations. Back to the EPCP example, there are three transitional periods (Fig. 3) and thus three as-is-and-to-be generation pairs: v1–v1.1, v1.1–v1.2 and v1.2–v2. Generation v1 denotes the current design of EPCP and is shown earlier in Fig. 1. The design for generation v2 is undertaken with conventional design activities. So it remains to derive generations v1.1 and v1.2. Given that change cases CC21, CC31

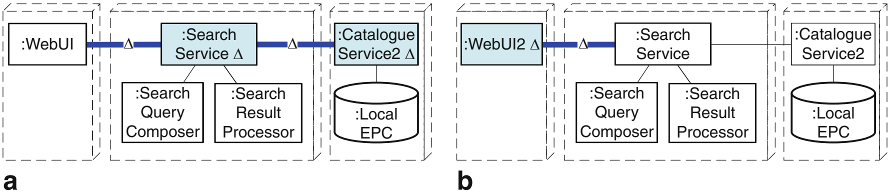


Fig. 4 EPCP: to-be generations after transitional periods a and b. **a** Generation v1.1 after transitional period a, **b** generation v1.2 after transitional period b

and CC11 add the i18n capability to EPCP, determining these generations becomes straightforward: affected transformable items (i.e. Catalogue Service and WebUI) are upgraded/extended. Consequently, these generations are depicted in Fig. 4, with changes from previous generations highlighted and labeled with the delta symbol "Δ".

Once the as-is and to-be generations are determined, the change case set is refined to uncover any potential change cases which could not be determined during the Application Lifecycle Analysis process. A rational for refining the set is that once a change is proposed for an application, the effect of the modification may ripple through other parts of the application (identified with change impact analysis [3]). Therefore, additional change cases may be required to also account for parts also affected by the proposed change. Take the case of "CC11: Add i18n support to Catalogue Service". An analysis reveals that since Catalogue Service will provide pricing information in different currencies and product descriptions in different languages, Search Query Composer must be able to take the currency unit as a parameter when specifying a price range for a search. Similarly, Search Result Processor will also be modified. This results in two new change cases:

- CC22: Add i18n support to Search Query Composer
- CC23: Add i18n support to Search Result Processor

Consequently, the runtime structure for generation v1.1 is updated to accommodate these two change cases. This same step is performed for the other change cases (CC11 and CC32) but yields no further change cases. Accordingly, the resultant generation v2 is now finalized as in Fig. 5 with changes since generation v1.2 highlighted and labeled with "Δ".

After change case refinement, transformations can be identified. For simplicity and illustration only, each change case will be realized by one or more transformations as listed in Table 2. Note that an extra transformation called "WebUI reconfiguration" is

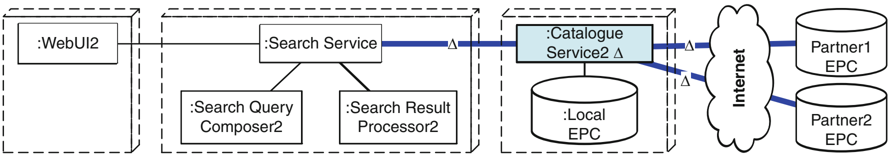


Fig. 5 EPCP: generation v2 after transitional period c

Table 2 EPCP: responsible transformations for refined change cases

Change case	Responsible transformation(s)	Transitional period
CC31: Add i18n support to Catalogue Service	Catalogue Service2 deployment, Catalogue Service removal	a
CC21: Modify Search Service to integrate with new Catalogue Service	Search Service reconfiguration	a
CC22 Add i18n support to Search Query Composer	Search Query Composer2 deployment, Search Query Composer removal	a
CC23: Add i18n support to Search Result Processor	Search Result Processor2 deployment, Search Result Processor removal	a
CC11: Add i18n support to WebUI	WebUI2 deployment, WebUI and WebUI2 reconfiguration, WebUI removal	b
C32: Modify Catalogue Service to integrate with two external EPC systems	Catalogue Service2 reconfiguration	c

added for change case CC11 to handle existing web (browser) sessions with WebUI. It is further elaborated during Transformation Design.

4.3 Transformation Design

This process aims to produce the detailed design for a transformation. It also identifies the changes to the zone(s), say, to accommodate new transformable items, and specifies how transformable items affected by the transformation will offer their functions or services during and/or after the transformation. To demonstrate, consider the transformations listed in Table 2 which can be summarized into three types: deployment, removal and reconfiguration. For a deployment transformation, the design is straightforward as it does not interrupt to the normal operations of EPCP. The following repeatable atomic steps can be undertaken:

1. Acquire adequate resources for X from the hosting zone.
2. Create X in the hosting zone.
3. Initialize X to a start-up state.
4. Bind X to appropriate transformable items.

Likewise, the design for a removal transformation follows a transformation pattern suitable for all cases in EPCP. Except the first step, the sequence below has the reverse effect of the steps in a deployment transformation:

1. Disable access to X.
2. Unbind X from the rest of the application.
3. Set X to a shutdown state.
4. Destroy X in the hosting zone.
5. Release resources allocated to X to the hosting zone.

On the other hand, thought must be given to the three reconfiguration transformations since they serve slightly different purposes. To begin with, the “Catalogue Service2 Reconfiguration” transformation aims to reconfigure the new version of Catalogue Service to bind to the two external EPC systems. For data integrity, Catalogue Service2 will need to be temporarily out of service while being bound to these two EPC systems. A simple strategy is to momentarily queue incoming search requests to it, and hence the following sequence:

1. Wait for Catalogue Service2 to complete outstanding search requests.
2. Block and queue incoming requests to Catalogue Service2.
3. Bind Catalogue Service2 to two EPC systems.
4. Unblock incoming requests for Catalogue Service2.

Similarly, the “Search Service reconfiguration” transformation can adopt the same strategy but with an extra step before step 3 above to unbind Search Service from the old version of Search Query Composer and Search Result Processor before binding Search Service to the new version.

Lastly, the “WebUI and WebUI2 Reconfiguration” transformation is more complicated. Given that at any time there may still be end users accessing EPCP, and hence the presence of several web sessions. It is not ideal to switch EPCP over to the new user interface (i.e. WebUI2) instantly for end users may be confused with the spontaneity of the change in the user interface. A suitable strategy is to keep WebUI handling existing web sessions, whilst all new and future web sessions are dealt with by WebUI2. Over time, all sessions served by WebUI will time out or be closed by end users. After that, WebUI is no longer used and can be removed from EPCP (to be handled by a removal transformation). Therefore, a delegation strategy is useful to this transformation:

- Redirect all incoming requests from new web sessions to WebUI2. The delegation mechanism can be implemented in the web container which hosts the old and new versions of the WebUI.
- Wait for all web sessions served by WebUI to close or expire.
- Disable access to WebUI.

4.4 Transformation Agent Design

The Transformation Agent Design process identifies the transformation agents to progress a particular generation (i.e. as-is) of an application to its next one (i.e. to-be) during a transitional period, their responsibilities in terms of which transformations they will perform, and the order in which they will perform the assigned transformations. Continuing on the EPCP example, the first step is to allocate transformation agents in the architecture. A logical way is to put a transformation agent in each of the zones (Fig. 1) to handle transformations within that zone: Web Agent, Engine Agent and Repository agent.

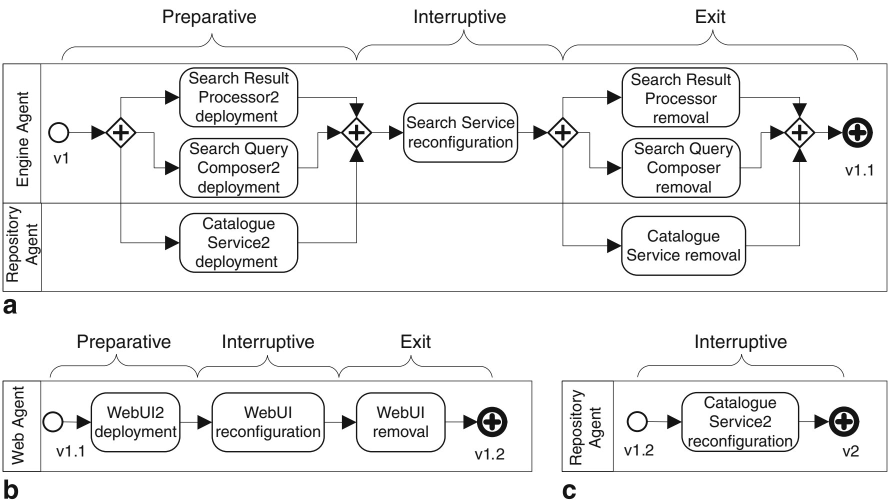


Fig. 6 EPCP: orchestration designs for transformation agents. **a** Transitional period a, **b** transitional period **b**, and **c** transitional period c

Next, transformations in each transitional period are assigned to transformation agents and arranged into an orchestration which specifies some kind of order in which the transformations will be performed, and can be represented using the Business Processing Modeling Notations [13] with custom enhancements. In this regard, a transitional period is split into three phases if applicable, to factor out transformations that will not interrupt an application from those which will, to confine the window of interruption. Consider transitional period a (Fig. 6a). During the *preparative* phase, new versions of Search Result Processor, Search Query Composer and Catalogue Service are set up ready for use in EPCP since their respective transformations will not affect the running of EPCP. During the *interruptive* phase, Search Service will be temporarily out of service to allow for it to bind to these new transformable items—leading to interruptions to end users. Finally, in the *exit* phase, unused and old versions of these three transformable items can be removed from EPCP since their respective transformations do not affect EPCP. A similar design heuristics is applied to transitional periods b and c, resulting in the orchestrations shown in Fig. 6b, c respectively.

5 Preliminary Evaluation

As a first step toward Continuum validation, we undertook an expert review as a non-empirical assessment of the initial version of Continuum. Experts were given the documentation of Continuum and specifically requested to comment on its

Table 3 Kinds of refinements for continuum based on expert view feedback

Area	Refinements
Concepts	More suitable terms, clearer concept definitions, new concepts
Processes	More suitable terms, elimination of overlapping features among processes, extension of processes to handle situations not originally dealt with
Process steps & elements	Clearer and more coherent steps, improved sequencing of steps, clearer linkage from elements to analysis and/or design
Modeling	New notations
Documentation	Improved readability and navigation, improved examples (q.e. EPCP)

strengths, weaknesses and any suggestions for further improvement. The documentation comprised an introduction which is an expanded version of this paper and the full specifications of Continuum. One expert was from academia, and another from the IT industry, took part in this review. The former did research in dynamic adaptation, while the latter had experience in dealing with architectural aspects and implementation of live upgrade to applications. An advantage of having experts from different sectors is that their knowledge and experience complement each other and potentially improved the comprehensiveness of the review results. Refinements were made to Continuum, as summarized in Table 3, according to the review results and subsequently verified by the experts.

We further evaluated the feasibility of incorporating Continuum into a tool and then a methodology, by firstly developing a methodology plug-in comprising its process elements using the Eclipse Process Framework (EPF), an open-source tool for methodology construction and tailoring [15]. Afterwards, the plug-in was incorporated into OpenUP, a small methodology also built with EPF [15]. A sample process lifecycle, based on Fig. 2, was then instantiated from OpenUP extended with the plug-in for a hypothetical project. The implementation results were used to further improve the structure of Continuum and how its elements link together. For instance, coupling among processes were spotted and reduced. The sequence and clarity of process steps were also improved to make them explicit.

6 Conclusions and Future Work

A process-oriented approach called Continuum has been described to embrace dynamic evolution in distributed applications utilizing the idiosyncrasies of a composition-based paradigm. It emphasizes application lifecycle, transitional period and transformation. Four Continuum processes have been presented to show how dynamic changes to a running application can be analyzed and designed. Continuum has gone through an expert review and a trial implementation into a tool after which refinements were made. Our next step involves using Continuum to analyze and design dynamic evolution for a commercial application to evaluate its completeness, usefulness and usability.

References

1. Arnold K, Gosling J, Holmes D (2005) The Java(TM) programming language, 4th edn. Addison-Wesley, Reading
2. Bennett KH, Rajlich VT (2000) Software maintenance and evolution: a roadmap. In: Proceedings of the 22nd international conference software engineering: the future of software engineerings. ACM Press, New York, pp 73–87
3. Bohner SA (2002) Extending software change impact analysis into COTS components. In: Proceedings of the 27th annual NASA Goddard software engineering workshop (SEW-27'02), Greenbelt, Maryland, pp 175–182
4. Brinkkemper S (1996) Method engineering: engineering of information systems development methods and tools. Inf Softw Technol 38(4):275–280
5. Ecklund EF, Delcambre LML, Freiling MJ (1996) Change cases: use cases that identify future requirements. SIGPLAN Not 31(10):342–358
6. ESI (2008) Welcome to ICCBSS 2008: weaving composite systems. http://www.iccbss.org/2008/. Accessed 3 June 2008
7. Evans H, Dickman P (1999) Zones, contracts and absorbing change: an approach to software evolution. In: Proceedings of the 14th ACM SIGPLAN conference on object-oriented programming, systems, languages, and applications (OOPSLA'99), ACM Press, New York, pp 415–434
8. Fung KH, Low G (2009) A methodology evaluation framework for dynamic evolution in composition-based distributed applications. J Syst Softw 82(12):1950–1965
9. Fung KH, Low GC, Ray PK (2004) Embracing dynamic evolution in distributed systems. IEEE Softw 21(2):49–55
10. ISO/IEC (2007) Software engineering—metamodel for development methodologies ISO/IEC 24744:2007. ISO/IEC Press, Geneva
11. Jayaswal K (2005) No time for down time. In: Jayaswal K (ed) Administering data centers: servers, storage, and voice over IP. Wiley, New York
12. Murata T (1989) Petri nets: properties, analysis and applications. Proc IEEE 77(4):541–580
13. OMG (2009) Business Process Modeling Notation Specification v1.2.
14. Oreizy P, Medvidovic N, Taylor RN (2008) Runtime software adaptation: framework, approaches, and styles. In: Companion of the 30th international conference on software engineering, Leipzig, Germany, pp 899–910
15. The Eclipse Foundation (2009) Eclipse Process Framework Project (EPF). http://www.eclipse.org/epf/. Accessed 7 Nov 2009

Regulated Open Multi-Agent Systems Based on Contracts

Emilia Garcia, Adriana Giret and Vicente Botti

Abstract Regulated Open multi-agent systems are composed by heterogeneous and autonomous agents which may need to coexist in a complex social and legal frame work that can evolve to address the different and often conflicting objectives of the many stakeholders involved. Of the ways in which agent behaviour can be regulated in a multi-agent system, contracts seem to be the most appropriate for industrial environments. This paper analyzes the open challenges in the development of these kind of systems. It also proposes an architecture and a designing method which face up to these challenges by means of the integration of open multi-agent systems, service-oriented architectures, e-contracting and regulation enforcement mechanisms.

1 Introduction

Nowadays service-oriented computing has become the enabler technology for today open information systems for enterprises. In todays bergaining scenarios, the e-Business approach is becoming more and more a "must-have" tool. Over recent years, several works have focused on solving the problem of integrating the multiagent system and the service-oriented computing paradigms in order to model autonomous and heterogeneous computational entities in dynamic, open environments [5]. In order to adapt these kind of systems to industrial environments the agent social relationships, organizational behaviour, agent interactions and service interchanges must be regulated. Among the mechanisms for agent behaviour regulation in an open multi-agent system, electronic contracting seem to be the most suitable for industrial applications. Contracts allow businesses to operate with expectations of the behaviour of others based on high-level behavioural commitments, and provide flexibility in how they fulfill their own obligations [11]. Furthermore, contracts allow to integrate the top-down specification of organizational structures with the autonomy of participating agents [3]. They can describe the rights and responsibilities that

E. Garcia (✉)
Universidad Politecnica de Valencia, Valencia, Spain
e-mail: mgarcia@dsic.upv.es

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_20,

an agent for example, acquires when entering into an organization. Besides, the structure of the organization can be detailed by means of the formal description of the social relationships between agents. Despite a range of formalisms for expressing contracts in a multi-agent environment are emerging [3, 9], there is no complete approach that integrates the notion of contracts as a key element in the development of open regulated multi-agent systems (ROMAS) which covers the whole development process.

The main objective of our work is to provide a complete set of methods and tools that guide and help developers from the requirement analysis to the implementation and execution stage. In this paper, we analyze the open challenges in the development of this kind of systems (Sect. 2). The paper is focused on the definition of an architecture (Sect. 3) and a model (Sect. 4) that deal with the development of these kind of systems.

2 Challenges in the Field

Regulated Open multi-agent systems (ROMAS) are systems in which heterogeneous and autonomous agents may need to coexist in a complex social and legal framework that can evolve to address the different and often conflicting objectives of the many stakeholders involved. In our view there are basically five issues to be solved in order to successfully design ROMAS:

Social Structure and Coordination: Open MAS need to be specified from a high abstraction level (organizations and their structure) to the individual definition of each agent. Moreover, Open MAS should allow heterogenous agents and organizations to interact between them in order to achieve their social and individual goals. Currently there is a trend that implements these interchanges using services. This fact improves the interoperability and standardization of the communications. These types of systems are called Service-oriented Open multiagent Systems [5].

Norms: The most common way to regulate a system is by means of norms. Norms are usually used to define which behaviour is allowed in the organization, which is prohibited and which are the obligations of each agent. Nowadays there are several languages to define norms in a formal way [9, 1], but controlling that each agent follows the norms is still an open research topic.

Contracts: In ROMAS specialized literature commitments are formally defined as contracts. They can be classified into two types: contractual agreements and social contracts [3]. Contractual agreements defines the commitments between agents. They formally specify the interchange of services and products and the related norms. On the other hand, social contracts specify the social relationship between agents and organizations, i.e., which are the agents rights and duties inside the organization.

Validation/Verification: Another open topic is how the consistency and coherence of norms and contracts can be automatically checked inside an organization. The major part of research in this topic is focused on offline verification of norms by means of model checking [12]. In dynamic environments, the normative context of an organization may changes at runtime. New norms are added and new contracts are signed. Despite that, most of the proposals do not take into account this dinamicity and do not offer any control mechanism to verify the stability and coherence of the system. Model checkers could be applied at runtime offering more dynamicity, but in order to do so an specific multi-agent system architecture is required in order to capture and verify the normative context of the organization at runtime.

Controlling Mechanism: There are two main controlling mechanism for norms: regimentation and enforcement mechanisms. Regimentation mechanisms organize the system in such a way that external actions leading to a violation of those norms are made impossible. One example of this architecture is the implementation of Electronic Institutions by means of the Ameli platform [4]. Each agent is associated with an entity called 'governor' who filters and controls the activity and the communication of this agent inside the institution. This type of mechanisms have some drawbacks: (1) they drastically decrease agent autonomy; (2) difficult the interaction with external agents; (3) decrease the reactive time of the agents due to the control management; (4) it makes the organization and structure rigid. On the other hand, enforcement is based on the idea of responding after and when the violation of the norms has occurred. Such a response, which includes sanctions, aims to return the system to an optimal state. This approach is more flexible and respects agents' autonomy. Most of enforcement approaches are based on the fact that the actions that violate norms are observable by the system. This can be easily implemented for organizational norms [2], i.e., for the norms that are related with the behaviour inside the organization. But in open regulated systems where internal and external heterogeneous agents interact, it is very difficult to control whether the commitments between agents are fulfilled. The control of the fulfilment of norms outside the organization is an open research topic.

3 ROMAS Architecture

In this section we deal with the development of ROMAS. Our architecture is based on organizational MAS in which agents interact by means of services and where contracts are used to formalize the commitments between agents and organizations. In our approach the normative and contract management follow an enforcement mechanism mainly based on the complaints of agents when they believe that a norm violation has happened. Moreover, we offer a multi-agent architecture that is prepared to verify and control the coherence and consistency of the normative context.

Social Structure and Coordination: Our proposal is based on a Service-oriented open MAS architecture [5]. To this end, we work on models for engineering large-scale open systems in which the constituent entities interact among them by means

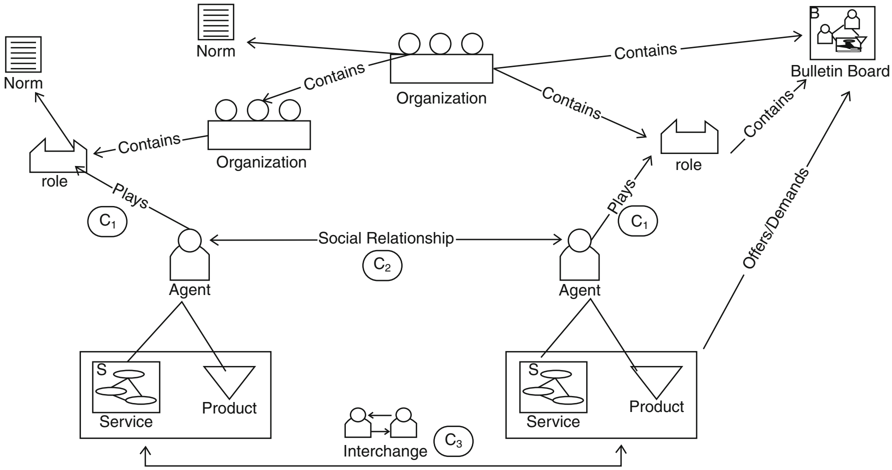


Fig. 1 Summarized ROMAS architecture

of services and whose relationships are formalized using contracts. The social structure that we propose is built upon *Virtual Organizations,* which represent a set of individuals and institutions that need to coordinate resources and services across institutional boundaries (Fig. 1). Virtual organizations are conceived as an effective mechanism for ensuring coordination and global goal fulfilment of an open system, in which heterogeneous entities (agents or services) interact and might also present self-interested behaviours. Moreover, service oriented architecture improves the interoperability, flexibility and modularity of agents.

Norms: Organizations are conceived as an effective mechanism for imposing not only structural restrictions on their relationships, but also normative restrictions on their behaviour. Figure 1 shows two types of norms: the norms associated with an organization that must be followed by each member of the organization; and the norms associated with a role that must be followed by the agents who play that role. Our approach to represent and implement norms is based on [2]. This model defines norms that control agent behaviours, the formation of groups of agents, the global goals pursued by these groups and the relationships between entities and their environment. The main features of this model are: (1) it covers the different norm levels which have a social meaning, i.e. interaction and institutional levels; (2) different types of norms are considered, i.e. constitutive, regulative and procedural norms; (3) it allows a structured definition of the normative system; and (4) it offers mechanisms for the creation and derogation of norms, by means of constitutive norms which define the dynamics of the legal system.

Contracts: In order to work in an open regulated system the commitments derived from negotiation should be formalized. In our approach, we analyze two types of contracts between agents: *Social contracts* and *Contractual agreements. Social contracts* can be defined as a statement of intent that regulates behaviour among organizations and individuals. As shown in Fig. 1, social contracts come up when an

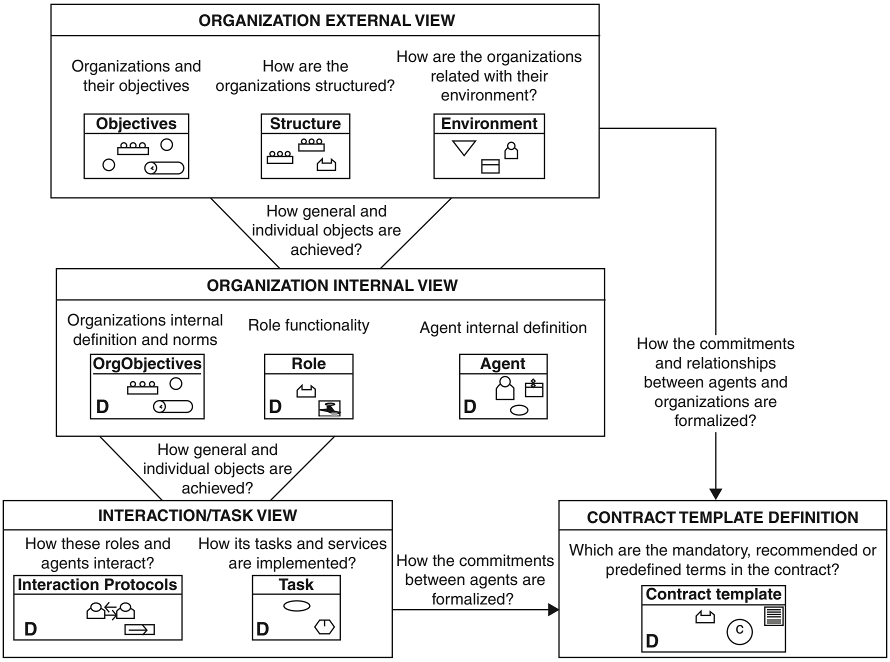


Fig. 2 Designing process

agent enters into an organization playing a role inside it (Fig. 1). Moreover, a social contract can be defined in order to formalize the social relationship between two agents (Fig. 2). Social order, thus, emerges from the negotiation of contracts about the rights and duties of participants, rather than being given in advance. *Contractual agreements* represent the commitments between several entities in order to formalize an interchange of services or products (Fig. 3).

Trading Process: Offers and demands of services and products could be publicly available by means of their publication in the *Bulletin Board* (Fig. 1). The *Contractual agreements* can be considered as an information artifact for Open MAS. This artifact facilitates trading processes because when an agent needs a service, it can consult the *Contractual agreements* to check if someone offers this service. In that case a prenegotiation process begins. During the prenegotiation stage agents should establish a common ontology which clarifies the semantic of the attributes of the negotiation. If they cannot found a common ontology, they must use a mediator in order to work as an interpreter between them. During the negotiation stage, agents can argue following specific protocols. There are many works which specify negotiation protocols [7, 8], furthermore, our approach offers the designer the mechanism to define their own interaction and negotiation protocols. The result of the negotiation is a contract proposal agreed by all the agents involved in the contract. This proposal must be validated by the organization. A validation process is required in

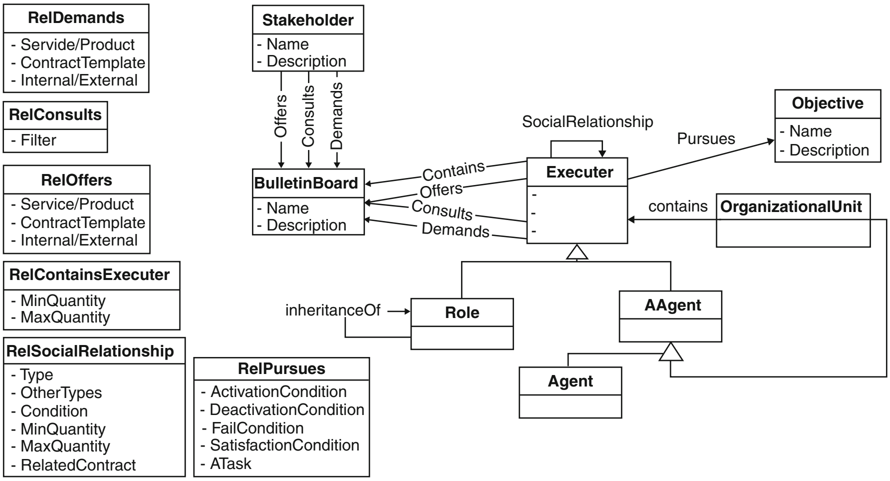


Fig. 3 Organization external view

order to ensure that the contract is correctly described, if it follows the normative context and if its clauses contradict or not previous signed contracts. If the validation finalizes successfully, the agents will sign the contract and the organization will save this contract in its knowledge base for taking it into account in future validations.

Executing a Contract: A contract execution usually implies the interchange of products and services among agents. In order to formalize this interchange, when an agent receive a product or service, it sign a *Receipt* to the provider agent. Our approach implements an enforcement mechanism to check norms and contract compliance. Normative and contractual management are implemented simulating real life human societies. Only few critical norms are controlled by the organization, whereas most violation detection is delegated to the victims. For example, when a sold product has not been transferred to the customer, the customer can fill a *Complaint* notifying it to the normative system. The organization should validate or reject the complaint taking into account the presented *Receipts* and any other proves and/or allegations. If the violation is proved the organization will sanction the corresponding agents. Alternative Dispute Resolution (ADR) or Online Dispute Resolution (ODR) mechanisms can be applied at this stage [10].

4 Designing ROMAS

ROMAS systems are dynamic and complex and requires to integrate different approaches and technologies. Therefore, a new designing method for developing these kind of systems is necessary. In this section, we present an overall view of the development process when engineering ROMAS systems. Then the meta-model for this development process is described. It provides support for the architecture presented in Sect. 3. Both the meta-model and the development process are based on a previous work [5] which defines a Service-oriented Open Multi-agent system modeling environment. Therefore and due to space limitation, this section is focused on the design of the contract notion and how it can be integrated in the different parts of the system. Further details about other concepts and model entities can be consulted in [5].

4.1 Development Process

The first process stage is the analysis of the system requirements, in order to define the global goals of the organization and the functionality that the organization offers and demands (Fig. 2). Next, the components of the organization are defined, i.e. the Organizational Units (OU), which represent groups of members of the organization; the roles defined inside each OU that will be related to the system functionality; their social relationships. All this is depicted in the **Organization external view**.

Secondly, the analysis of the goals of the organization is carried out in which the global goal of the organization is refined into more specific goals, which represent both functional and non-functional requirements that should be accomplished by the organizational units of the system. This analysis and the individual objectives of each entity should guide the internal definition of them. Furthermore, global and individual objectives will determine the norms that control the global behavior of the OU members. The internal definition of the OUs is defined, by means of different instances of the **Organization internal view**, one for each OU, agent and role.

The way in which the OU entities achieve their goals is defined by means of the **Interaction/Task view**. This meta-model view allows defining both interaction protocols and the sequence of activities in which a task is decomposed.

Social and contractual commitments can be formalized using contracts. Despite that contracts are defined at runtime after the negotiation stage, designers can specify, at design time, which protocols should follow a contract to be formalized. Also they can define some features and norms that the final contract must fulfill. To this end, it is used the **Contract Template view** entity.

This is not a linear process but an iterative one, in which the identification of a new functionality element of the system implies the integration of this element in all the diagrams into which it is needed.

4.2 Meta-Model

As is explained in the introduction of this section, this meta-model is based on [5] so only the new parts, which are related with the contract notion, will be explained here.

The **Interaction/Task view** has no fundamental changes from the previous model, therefore, it is not detailed in this section and it can be consulted in the bibliography.

Organization External View: In this view the global goals of the organizations and the functionality that organizations provide and require from their environment are defined. The static components of the organization, i.e. all elements that are independent of the final executing entities are defined too. The part of the unified meta-model related with this view is presented in Fig. 3. Organizations, roles, agents and objectives are classical entities of organizational multi-agent system, so a full definition of its entities and its relationships can be consulted in [5]. The relation *Social Relationships* represents the type of social relationship between two entities. It is related with their position in the structure of the organization (i.e. information, monitoring, supervision). Some social relationships can have a *ContractTemplate* associated which formalize some predefined commitments and rights that must be accepted or negotiated during the executiontime. Each *Contract Template* is defined using the Contract Template view. The *Stakeholders* interact with the organization by means of the publication of offers and demands of *Products* and *Services* in the *BulletinBoard.* Each offer or demand published into the BulletinBoard can be associated with a ContractTemplate. It means that this offer or demand has some predefined restrictions which are specified in this ContractTemplate view. This mechanism allows the designer to define the interaction with external entities.

Internal View: This view allows defining the internal functionality, capabilities, believes and objectives of each entity (organizations, agents and roles) by means of different instances of this model (Fig. 4). Agents and even Organizations can play

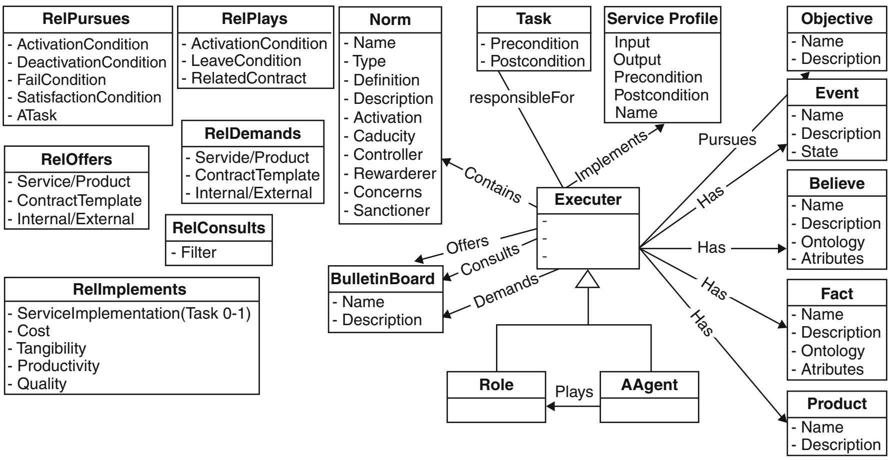


Fig. 4 Internal view

specific Roles inside another organization. Conditions for acquiring and/or leaving the role can be defined. *ActivationCondition* and *LeaveCondition* attributes of this *play* relationship indicate in which situation an agent can acquire or leave a role. Moreover, A *Related Contract* can be defined in order to specify which rights and duties will acquire this entity when it plays this role. The final contract will be defined at runtime, but some design decision can be represented by means of a Contract Template. Internal entities can publish offers and demands in a *BulletinBoard,* as external stakeholder can do by means of the organizational external view. This publications can also have an associated Contract Template to describe some predefined specifications. The relationship *Contains Norm* allows defining the rules of an organization and which norms are applied to each agent or role.

Contract Template Definition: This view allows defining *Contract Templates.* Contracts are dynamic entities which cannot be defined at the design stage. Despite this, designers could know some predefined restrictions that all final contract of a specific type should follow. These restrictions are defined in a Contract Template (Fig. 5). The relationship *Signants* indicates who is allowed to sign this type of contracts. It could be a specific agent, an agent who plays a specific role or an organization. A *ThirdPart* could be anyone who participates in the negotiation protocol or who is affected by the final execution of the Contract. The relationship *Protocol* indicates which protocols are recommended to negotiate this type of contract. After the negotiation, the *Notary* is responsible for verifying the correctness and coherence of the final contract definition. He should check if any term of a contract violate any norm of the regulated environment. Each type of contract can define which *Receipts* will be generated during the execution time. Receipts are proves of facts, for example, a receipt can be generated when an agent successfully provides a service. In case of conflict, the *Judge* has to evaluate the Complaints and the generated Receipts following the *ConflictResolution* protocol. If he decides that there has been a violation of a norm, the *RegulationAuthority,* who is the main authority in the context of a contract, can punish or reward the agent behaviours. The relationship *Hard terms* indicates that any instance of this type of contract has to include this norm. *Soft terms*

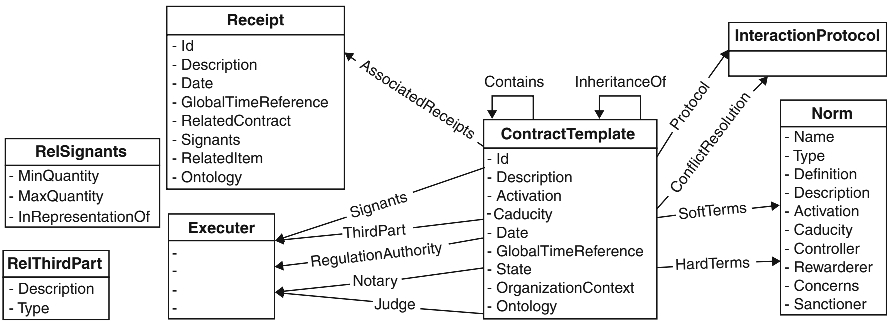


Fig. 5 Contract template definition

are recommendations, so during the negotiation stage *Signants* will decide whether this norm will be included or not in the final contract.

5 Using ROMAS

In order to illustrate the usage of the proposed meta-model and development process we have modeled a case study based on a water market, which is called mWater [6]. Let's suppose there is a water market that is an institutional, decentralized framework where users with water rights are allowed to voluntarily trade their water rights with other users, in exchange of some compensation, economic or not but always fulfilling some pre-established rules. It represents a virtual market base system, in which water right transfer agreements are executed by autonomous entities. They will be able to negotiate the terms and conditions of the transfer agreement following normative laws. Due to space limitations, only some simplified views of the case study are presented here.

Firstly, the main objectives of the *mWater organization,* its structure and the relation with its environment are defined by means of the Organizational external view (Fig. 6). The mWater market is modeled as a virtual organization that pursues the global goal of *Control the Market.* This organization is composed by *WaterUsers* who can be *Buyers* or *Sellers.* Moreover, some Buyers can be organized as *aBuyersCommunity.* There are two governing roles involved in grievances and contract validation processes: *BasinRegulatinAuthority* and *Jury.* Thirdly, the *MarketFacilitator* role represents those institutional agents who run standard trading activities for example set up a trading table or mediate in a face-to-face negotiation.

The role *MarketFacilitator* manages the *BulletinBoard* where agents and organizations can publish offers of Water Rights. The *BasinRegulatinAuthority* role has a social relationship with the *Jury* role, the first one has to inform the second one. The terms of this relationship are defined using the Contract Template View.

In this case study there is no service that can be accessed by an external agent because any agent in the market must be registered as a Water User, therefore no stakeholder has been defined.

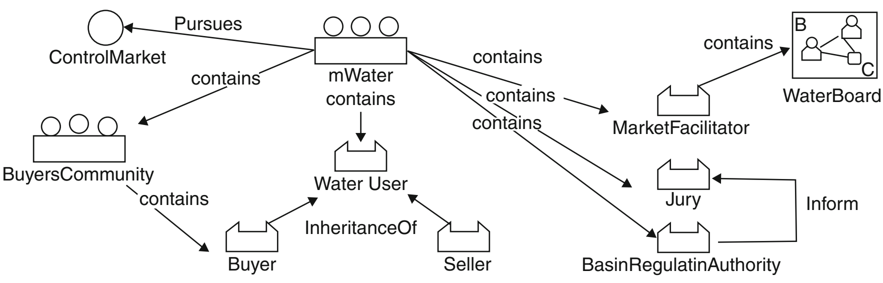


Fig. 6 mWater external view

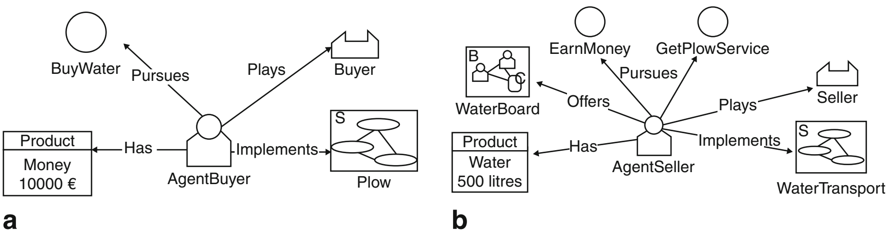


Fig. 7 **a** Agent buyer internal view. **b** Agent seller internal view

The next step in the development process is defining the internal functionality, capabilities, believes and objectives of each entity (organizations, agents and roles) by means of different instances of the **Internal view** model. Figure 7 shows two examples of this type of diagram. The first one is the *Agent Buyer* definition. This agent pursues the objective *BuyWater.* It implements the service *Plow* and has 10.000 eur. The second one is the *Agent Seller* definition. This agent pursues two objectives: *GetPlowService* and *EarnMoney.* It implements the service *Water transport* and has Water. The *Agent Seller* publishes an offer of water in the *Water Board Bulletin*.

Therefore, when the Agent Buyer tries to achieve their objective of *BuyWater,* it executes an interaction protocol which is defined by means of the Interaction/Task view. The final contract cannot be defined at design time, but the mWater organization establishes some restrictions that each transaction in this market should follow. This restrictions are represented in the Contract Template definition of Fig. 8.

The model shows that only agents who play the role of *Seller* and *Buyer* can sign the contract. Designers recommend the *BuyRight Protocol* in order to define completely this type of contracts. The agent who plays the role *MarketFacilitator* is the responsible for verifying the correctness and coherence of the final contract.

The agent who plays the role *Jury* will mediate and judge if there is any Complaint related to this contract. In order to solve the conflict, the protocol *Alternative Dispute Resolution* will be executed. Moreover, the contract template specifies that the agent who plays the role of *BasinRegulatinAuthority* will be main authority in the context of this contract.

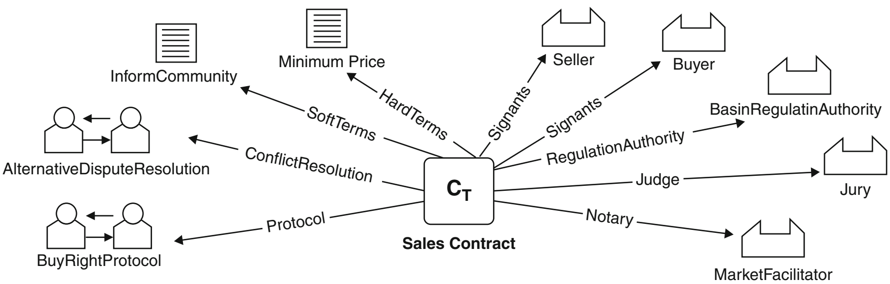


Fig. 8 mWater contract template definition

Every Sale Contract should include the norm *MinimumPrice*=(FORBIDDEN, price < 0.07 eur/liter), which means that it is forbidden to sell water for less than 0.07 eur/liter. The norm *InformCommunity*=(OBLIGATION, Seller, inform(mWaterOrganization)) , means that the Seller should inform the community about the final terms and conditions of the agreement. It is a *Soft Term,* i.e., it is only a recommendation. Therefore, during the negotiation process the signants will decided if this norm should be included or not in the final contract.

6 Conclusions and Future Work

In this paper, we have dealt with the problem of engineering complex systems composed by heterogeneous and autonomous entities which coexist in a complex social and legal framework that can evolve to address different conflicting objectives of many stakeholders. After the analysis of the most important features for developing these kind of systems Regulated Open Multi-agent Systems based on contracts seem to be the most suitable approach. A flexible architecture and a design process for developing these kind of systems has been presented and used to design a case study. Some of the benefits of this approach are: (1) Multi-agent technology provides a high level of abstraction models to simulate social structure and coordination. (2) Service-oriented technology provides an interoperable and standarized interaction mechanism. (3) Contracts can be linked with general social structures such as norms, social relationships and organizations. They can model functional and non-functional properties. Additionally, they allow integrating the top-down specification of organizational structures with the autonomy of the entities. (4) The enforcement architecture to control the satisfaction of norms and contracts respect the flexibility and autonomy of the agents' behaviour. Moreover, it could be used in open environments where external agents interacts with internal organization entities. Currently, we are still working on methods and tools for engineering ROMAS. Our next step is to apply model checking techniques in order to verify the coherence and consistency of contracts in regulated environments. We plan to integrate these algorithms into the presented model.

Acknowledgements This work is partially supported by the TIN2008-04446, TIN2009-13839C03-01, PROMETEO 2008/051 projects, CONSOLIDER-INGENIO 2010 under the grant CSD200700022 and the FPU grant AP2007-01276 awarded to Emilia Garcia.

References

1. Agotnes T, der Hoek WV, Rodriguez-Aguilar JA, Sierra C, Wooldridge M (2007) On the logic of normative systems. AAAI Press, Menlo Park, pp 1175–1180
2. Criado N, Argente E, Botti V (2009) A normative model for open agent organizations. In: ICAI, vol 1. CSREA Press, Las Vegas, pp 101–107

3. Dignum V, Meyer J, Dignum F, Weigand H (2003) Formal specification of interaction in agent societies. Formal Approaches to Agent-Based Systems (FAABS), p 2699
4. Esteva M, Rosell B, Rodriguez JA, Arcos JL (2004) AMELI: an agent-based middleware for electronic institutions. In: Proceedings of AAMAS04, pp 236–243
5. Garcia E, Argente E, Giret A (2009) A modeling tool for service-oriented open multiagent systems. In: PRIMA, vol 5925 of LNAI. Springer, New York, pp 345–360
6. Garrido A, Giret A, Noriega P (2009) mWater: a sandbox for agreement technologies. In: CCIA 2009, vol 202. IOS Press, The Netherlands, pp 252–261
7. Ito T, Klein M, Hattori H (2008) A multi-issue negotiation protocol among agents with nonlinear utility functions. Multiagent Grid Syst 4(1):67–83
8. Lai G, Sycara K, Li C (2008) A decentralized model for automated multiattribute negotiations with incomplete information and general utility functions. Multiagent Grid Syst 4:45–65
9. Oren N, Panagiotidi S, Vázquez-Salceda J, Modgil S, Luck M, Miles S (2009) Towards a formalisation of electronic contracting environments. In: Hübner J, Matson E, Boissier O, Dignum V (eds) Coordination, organizations, institutions and norms in agent systems IV. Lecture notes in computer science, vol 5428. Springer, Berlin, pp 156–171. doi: 10.1007/978-3-642-00443-8_11. http://dx.doi.org/10.1007/978-3-642-00443-8_11
10. Pitt J, Ramirez-Cano D, Kamara L, Neville B (2008) Alternative dispute resolution in virtual organizations. In: Artikis A, O'Hare G, Stathis K, Vouros G (eds) Engineering societies in the agents world VIII. Lecture notes in computer science, vol 4995. Springer, Berlin, pp 72–89. doi: 10.1007/978-3-540-87654-0_3. http://dx.doi.org/10.1007/978-3-540-87654-0_3
11. Vazquez-Salceda J, Confalonieri R, Gomez I, Storms P, Nick Kuijpers SP, Alvarez S (2009) Modelling contractually-bounded interactions in the car insurance domain. DIGIBIZ 2009
12. Vigano F, Colombetti M (2007) Symbolic model checking of institutions. In: ICEC, pp 35–44

Modeling Method for Bridging Pragmatic and Semantic Dimensions of Service Architectures

Prima Gustienė and Remigijus Gustas

Abstract Rapid changes in business environment result in necessity to introduce new agile modeling methods, which effectively support evolution of information system specifications. Service-oriented paradigm can be applied for achievement of this goal. Conceptual models of service interactions are important as a natural principle for the separation of crosscutting concerns of various information system architecture dimensions. This paper presents a modeling method for systems analysis and design, where service orientation is used for integration of information system conceptualizations. Goals, problems and opportunities represent pragmatic aspects, which are considered as a driving force of design process. It is a starting point for reasoning about changes of computation-neutral specifications of service architectures across organizational and technical system boundaries. The presented method is used for bridging pragmatic and semantic dimensions of information systems specifications.

1 Introduction

Unprecedented rates of changes in business processes and technological advances have made it more difficult for information systems development team to be agile in responding to changing requirements (Lee and Xia 2010). Various models and methods are used to support the process of information system (IS) analysis and design, but still after many years of practice, there are a lot of unsolved problems that cause system development projects to fail. One of the reasons is that rapid changes in business environment result in necessity to introduce new business solutions, which should be effectively supported by computerized information systems. Such situation increases complexity of IS specifications and impedes the

P. Gustienė · R. Gustas (✉)
Department of Information Systems, Karlstad University, Karlstad, Sweden
e-mail: prima.gustiene@kau.se

R. Gustas
e-mail: remigijus.gustas@kau.se

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_21,

alignment of business goals with computerized system design. System designers have difficulties to separate crosscutting concerns (Jacobson and Ng 2005) of evolving conceptualizations across various IS architecture dimensions (Zachman 1987). Another problem is that efficient methods for the integrated modeling of business and information technology (IT) services are still not available.

Service orientation is an architectural style that helps to analyze systems as loosely coupled compositions of subsystems. Service-oriented principles (Erl 2005) are bound to implementation-oriented system development style, which is known as service-oriented architecture (SOA). SOA is mostly related to IT related enterprise system part. Nevertheless, the notion of service goes well beyond services that are realized by software. Therefore, it is necessary to have a broader understanding of service concept in IS development. Conflicting views on the concept of service is one of the obstacles to the attempts to develop a new science of services (Chesbrough and Spohrer 2006) and new academic programs focusing on services (Alter 2008). The ontological foundation of service concept has no direct relation to technological solutions. It is service value that must be delivered. Value cannot be created without interactions between service requester and service provider.

A service can be defined by using a collection of ordered and purposeful interactions between human, organizational or technical components. Communication actions prescribe the responsibilities of actors involved. They take place between service requester and service provider. Every action is goal-driven and it should always result in some value flow (Gordijn et al. 2000) to the actor. Service from an IS analysis point of view can be defined (Gustas and Gustiene 2009) *by a number of interaction flows into opposite directions between a service requester and service provider. Each service response is a function of a service request.* Service providers are actors who receive service requests and transform them into responses, which are sent to service requesters. This idea is represented graphically in Fig. 1.

Service architecture can be conceptualized as composition of various types of service interactions that take place between enterprise actors. Any service interaction loop must be motivated by the resulting value flow. In this way, an enterprise system can be defined by using a set of interacting loosely coupled subsystems. Subsystems are viewed as technical or organizational components. Organizational components can be individuals, companies, divisions or roles, which denote groups of people. Technical components are subsystems that can be represented by software and hardware.

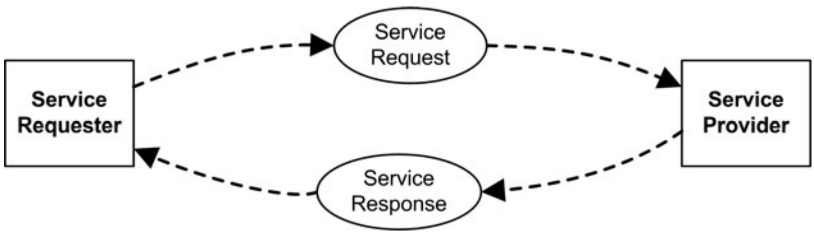


Fig. 1 Service interaction loop

The most fascinating idea about a service concept is that it can be applied equally well to organizational as well as technical settings. Therefore, computation-neutral representations of service architectures can be used for bridging a gap between business experts and system designers. In the public sector, services may denote organizational processes that are triggered in response to some unexpected events. According to some authors, services cannot be transferable, because they are events, not objects (Ferrario and Guarino 2008). Service interactions create and capture service value in an exchange between two or more parties. Currently, service construct is not used explicitly in the conventional information system analysis and design methods. As a consequence, service architectures are difficult to visualize and comprehend for business experts, who determine organizational strategies. IS development tradition tends to draw attention away from strategic business modeling aspects and concentrates on implementation-specific artifacts. Pragmatic aspects of services are left behind. It is crucial to demonstrate how the conceptualizations of service architectures are related to the purpose of the system. The goal of this paper is to briefly present a service-oriented modeling method, which is developed to challenge semantic integration and evolution problems of IS conceptualizations.

2 Pragmatic Dimension of Services

Pragmatic analysis clarifies the scope of possible changes in organization. Pragmatic dimension of services provides motivation behind new business strategies. The analysis of goals is important as they motivate and drive the overall information system analysis and design process. This is the first step, where goals are stated and discussed. Strategic choices taken in an early system analysis phase are a key in determining how IS changes will be approached (Rouse and Baba 2006). Pragmatic description is supposed to give a definition of the 'why'—a long term intention or a vision of information system under development. From the design point of view, the desired information system functionality can be conceptualized in terms of services, which are analyzed as actors' goals, problems and opportunities.

Enterprise system functionality can be characterized as desired or problematic. Desirable situations are motivated by opportunities or goals. Undesirable situations are expressed as problematic, which should be avoided. Pragmatic dependencies (Gustas and Gustiene 2008) can be used to analyze intentions of actors involved in business processes. Actor intentions are viewed as a driving force in designing or reasoning about interactive, behavioral and structural dimensions of IS architectures. Graphical notation of the pragmatic dependencies is presented in Fig. 2.

One important dimension of business process analysis is the ways in which different goals, opportunities and problems are related. Pragmatic descriptions also help IS designers to understand and motivate the ways in which the high-level objectives can be achieved. Pragmatic entities can be decomposed by refinement dependency into more specific elements. Figure 3 presents the example of goal decomposition.

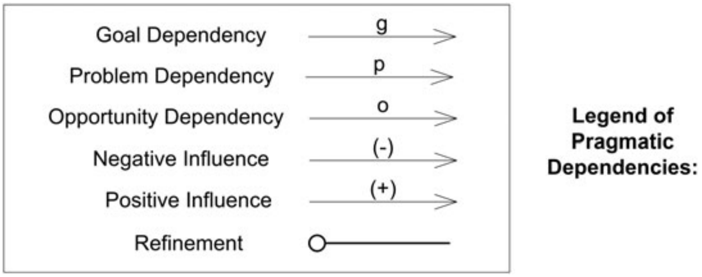


Fig. 2 Legend of pragmatic dependencies for service analysis

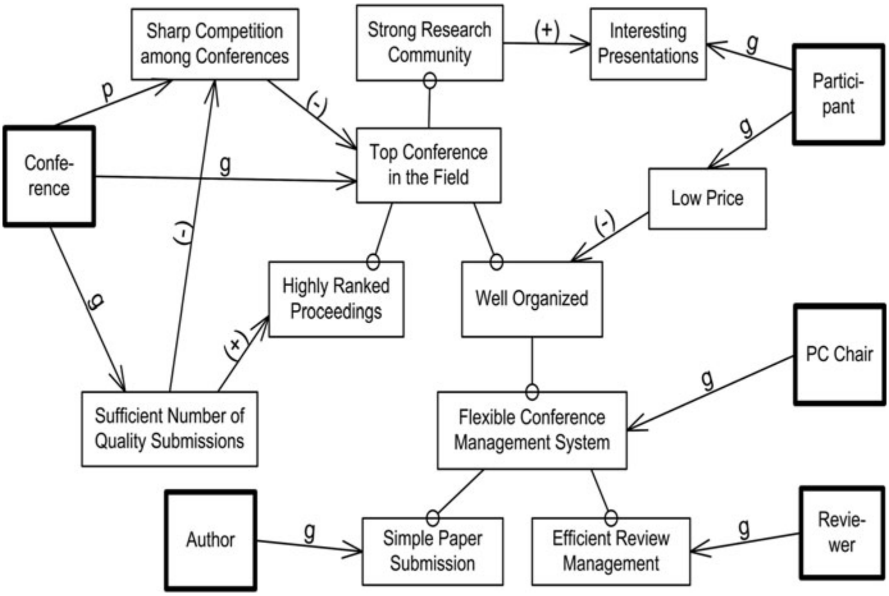


Fig. 3 Example of dependencies among various actors' goals

Problematic situations (such as 'Sharp Competition among Conferences') negatively affect reaching one of the goals 'Top Conference in the Field'. One way to eliminate a problematic situation is to weaken a problem by using some opportunity or by introducing a new goal (such as 'Sufficient Number of Quality Submissions'), which influences negatively a problem or it influences positively a goal. Another way to satisfy a goal is to decompose it into sub-goals, which can be implemented as various types of services. In the example above, three sub-goals support achievement of goal 'Top Conference in the Field'. The sub-goal 'Well Organized' can be partially achieved by 'Flexible Conference Management System', which is decomposed into two sub-goals 'Simple Paper Submission' and 'Efficient Review Management'.

Goal dependency is used to specify the desirable situations. It links enterprise actors with such intentional notions as goals or objectives. Problem dependency is used to specify undesirable situations. Enterprise actors intend to avoid them. Opportunity dependency can be used to refer to future situations that can be taken advantage to improve a current situation. Opportunities are strength of an existing enterprise system that should be maintained in a new system. For instance, some services can be viewed as opportunities. The desirable features of problematic situation can be indicated as opportunities. Negative influence dependency (−) and positive influence dependency (+) are used to denote influences among goals, problems and opportunities. Negative influence dependency from A to B indicates that A can be regarded as a problem, because it hinders the achievement of goal B. The positive influence dependency from A to B would mean that A can be viewed as an opportunity for the achievement of goal B. Refinement dependency is used as a means of pragmatic decomposition of goals, problems and opportunities. Pragmatic entities are typically refined by more specific entities, which represent the desired or problematic situations.

The pragmatic descriptions of services help business experts to reason about the possible changes for creation of an additional business value. Pragmatic aspects characterize IS services in a prescriptive way (Ferrario and Guarino 2008). Goal analysis is also useful for identifying conflicts, as the meeting of one goal may interfere with other goals. It should be noted that the interpretation of problem, opportunity and goal is relative. The achievement of one actor goal can be regarded as a problem for another actor. Pragmatic analysis is crucial, because the underlying assumption is that IS services are worthwhile if they meet goals of organization. The presented set of pragmatic dependencies can be viewed as an initial description of various actor intentions for constraining the development of IS specifications.

3 From Pragmatic Specification to Semantic Structure of Service Architectures

Pragmatic specifications aim to provide motivation for conceptual representations of enterprise components, which are defined in terms of interaction dependencies between service requesters and service providers. One of the benefits of service interactions is to analyze business processes in terms of service flows. The declarative nature of flows is very helpful in IS analysis phase, because they have very little to do with dependencies between business activities. The particular strength of interaction dependencies is possibility to capture crosscutting concerns (Jacobson and Ng 2005) among organizational components. Most conceptual modeling approaches do not take into account the notion of service flow, which demonstrates value exchange among actors involved in business processes (Gordijn et al. 2000).

The presented modeling method provides the way of bridging between semantic and pragmatic specifications. Pragmatic entities are refined into service-oriented diagrams (Gustas and Gustiene 2009). A small case study of a conference review management system will be used as a running example, which is important for the demonstration of conceptual modeling constructs and their expressive power. The initial description of 'Simple Paper Submission' and 'Efficient Review Management' can be described as follows: *One of the authors plays the role of* ***contact person*** *who* ***submits*** *a paper to a conference. The responsibility of a conference program committee (****PC****)* ***Chair*** *is to* ***appoint reviewers*** *for every submission.* ***Reviewer*** *is obliged to* ***return review*** *of the paper to the* ***PC chair*** *on time. Depending on the reviewing outcome, the* ***PC Chair*** *is authorized to* ***accept*** *or* ***reject*** *a submitted paper. If the paper is accepted, then revision instructions are sent to the corresponding* ***contact person****. Otherwise, reviewer comments are included in the rejection letter.* The refinement of two sub-goals into the described service interactions is represented in Fig. 4.

Conceptual models of interactions are not difficult to understand for business professionals as well as information system designers. Service interactions are helpful for clarifying why actors are willing to exchange business objects with each other. However, many conceptual modeling methods are not able to capture interaction flows between actors. Actions and flows can be viewed as the fundamental elements for defining business scenarios. A scenario is an excellent means for describing the order of interactions. Each interaction can be analyzed separately as is required by the principle of separation of crosscutting concerns. In such a way, service interactions provide a natural way of process decomposition. Service representations are built by conceptualizing interactions among organizational and technical components, which can be viewed as various types of enterprise actors.

Conceptual models should have capacity to describe various aspects of service architectures in a clear and sufficiently rich way. Parallel, sequential, exclusive or iterative behavior can be specified by using activity diagrams, but normally this type of specification is not associated with interactions (Glinz 2000), which are typically expressed by use case or sequence diagrams in the traditional modeling approaches

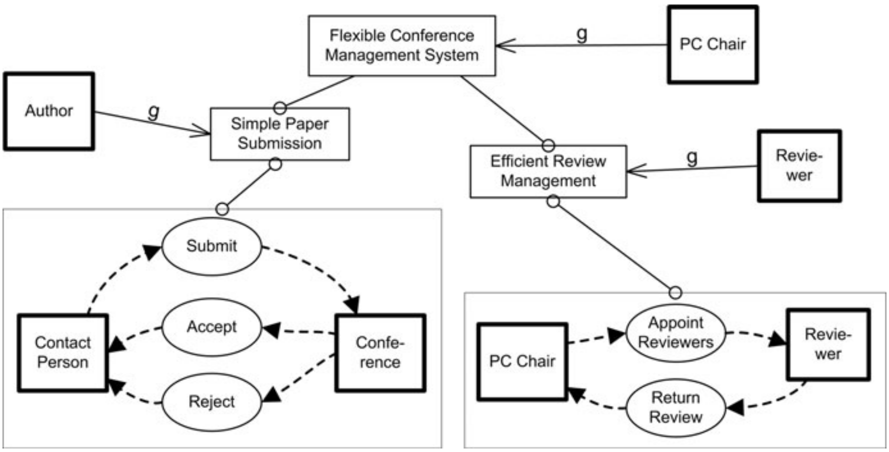


Fig. 4 Refinement of pragmatic entities in terms of service interactions

(OMG 2010). This situation creates difficulties in detection of discontinuities and breakdowns in IS specifications, because the knowledge on choreography of business interactions is missing. The static and dynamic aspects of service conceptualizations cannot be analyzed in isolation from the pragmatic specifications. Information systems methodologies are quite weak in integrating interaction flows and related effects of structural changes in various classes of objects.

The behavioral and structural aspects of interactions can be analyzed in terms of their reclassification, creation or termination effects (Gustas 2010). When two subsystems interact one may affect the state of each other (Evermann and Wand 2009). The structural changes of objects can be defined in terms of object properties. Interaction dependency R(A ---> B) between two subsystems A and B indicates that A is able to perform action R on one or more B subsystems. An action typically manipulates properties of some objects. Otherwise, this action is not purposeful. Property changes may trigger object transitions from one class to another. The internal changes of objects can be expressed by using transition links ($\longrightarrow$) between two object classes, which represent passive concepts. The graphical notation of reclassification construct is represented in Fig. 5.

Two kinds of fundamental changes occur in the presented reclassification action: removal of an object from a precondition class and creation of an object in a post-condition class. Reclassification construct with a missing post-condition class is used for representation of termination of a precondition class object. The construct without a pre-condition class represents creation of a postcondition class object. For example, *Submit* action can be defined as creation event and *Reject* action can be viewed as termination. *Appoint_Reviewers*, *Return_Review* and *Accept* are reclassification events (see Fig. 8). Object creation or reclassification without any properties does not make any sense. So, various types of static and dynamic dependencies between classes are used to define mandatory properties of objects. The lack of noteworthy difference between pre-condition and post condition class indicates that the specification of a communication action is either incomplete or a communication action is not useful. Pre-condition and post-condition classes are typically characterized by two different sets of mandatory attributes, which are sufficient for the representation of permissible ways, in which changes may occur. Static dependencies such as

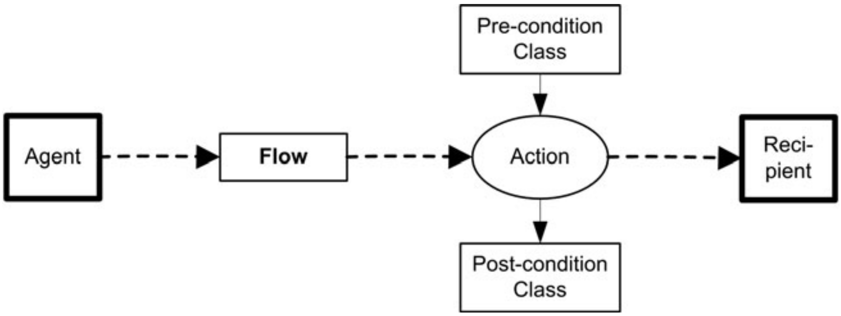


Fig. 5 Construct for representation of reclassification event

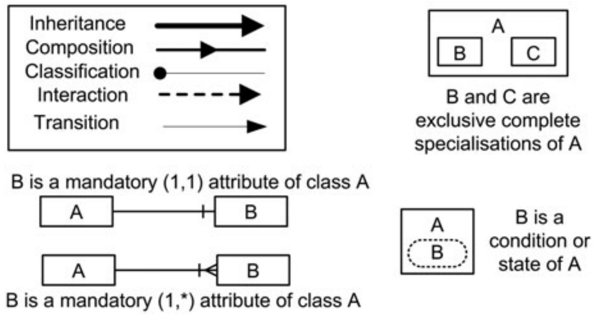


Fig. 6 Graphical notation of dependencies between concepts

inheritance, composition, single-valued and multi-valued mandatory attributes are sufficient to visually recognize and comprehend the details of various interaction effects. The graphical notation of concept dependencies is presented in Fig. 6.

One significant difference of our integrated modeling approach from the traditional ones is that all dependencies are nameless. Any concept can also be decomposed or specialized by using inheritance, classification and composition dependencies (Gustas 2010). A concept can be specialized by using a special conditions or state. The lifecycle of an object is typically represented by an initial, intermediate and final states (see Fig. 8). A creation event corresponds to a starting point and removal action—to the end point in object's lifecycle. The semantic integrity of static and dynamic aspects is crucial in defining interaction effects. It is not sufficient to represent what type of objects are created and terminated. Service architectures must clearly define the structural object transition effects in terms of attribute values, which must be either removed or preserved in any creation, termination and reclassification action.

4 Process of Service-Oriented Analysis

One of the main contributions of this paper is to explain seven fundamental steps, which support the incremental, systematic and computation-neutral service-oriented IS analysis process. These steps deal with both the semantic and pragmatic dimensions of service conceptualizations. The presented modeling process is crucial for the integration of interactive, behavioral and structural aspects of IS specifications. A starting point of service-oriented analysis is identification of actor goals and interaction dependencies among service requesters and service providers. The structural aspects represent business data. The behavioral aspects are clarified by defining object transition effects. Without ability to represent noteworthy structural changes, it would be difficult to understand the deep semantics of interactions. Possibility to reiterate seven modeling steps for various goals helps to integrate the static and dynamic aspects of IS conceptualizations. The service-oriented modeling process steps are as follows:

1. Identification of actors' problems, goals and opportunities. Pragmatic specification, which is illustrating the outcome of this step, is represented in Fig. 3. This specification demonstrates problematic and desired situations with respect to various organizational components. It also indicates influence and refinement links between various goals, problems and opportunities.

2. Refinement of goals, problems and opportunities in terms of service interactions. Two service interaction loops of a conference management system are illustrated in Fig. 4.

3. Identification of interaction flows and static dependencies among actors. Interaction flows are the special types of concepts that represent moving things. In our modeling approach, solid rectangles are used for the denotation of material flows and light boxes show data flows. An action with a missing data or material flow is understood as a decision or control flow. Actions are performed by actors and are represented by ellipses. They are necessary for transferring flows between subsystems, which are represented by various organizational components. Actors are denoted by square rectangles. The composition, inheritance dependencies together with the interaction flows among actors are illustrated in Fig. 7.

A contact person has a possibility to *submit* a paper. If submission is accepted, the responsibility of the conference PC chair is to trigger the *appoint reviewers* action, which is used to send review documents to reviewers. Reviewer is obliged to deliver review to PC chair by triggering the *return review* action. The PC Chair is authorized to either *accept* or *reject* a submitted paper by informing a contact person with a corresponding rejection letter or revision instructions.

4. Definition of actions in terms of transition dependencies. The internal effects of objects can be expressed by using transition links (→) between various classes of objects. There are three fundamental ways for representing object behavior by using reclassification, creation and termination actions (Gustas and Gustiene 2009). If termination and creation action is performed at the same time, then it is called a reclassification action. The graphical examples of creation and termination are presented in Fig. 8. For instance, a Submitted Paper is stored by using Submit action. A Reviewed Paper can be terminated by the Reject action.

5. Identification of a noteworthy semantic difference in every action. This step is important for identification of attributes, which are affected during object transitions

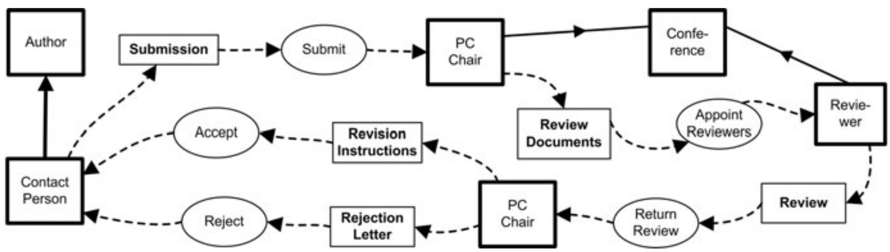


Fig. 7 Main actors and interactions in a conference management system

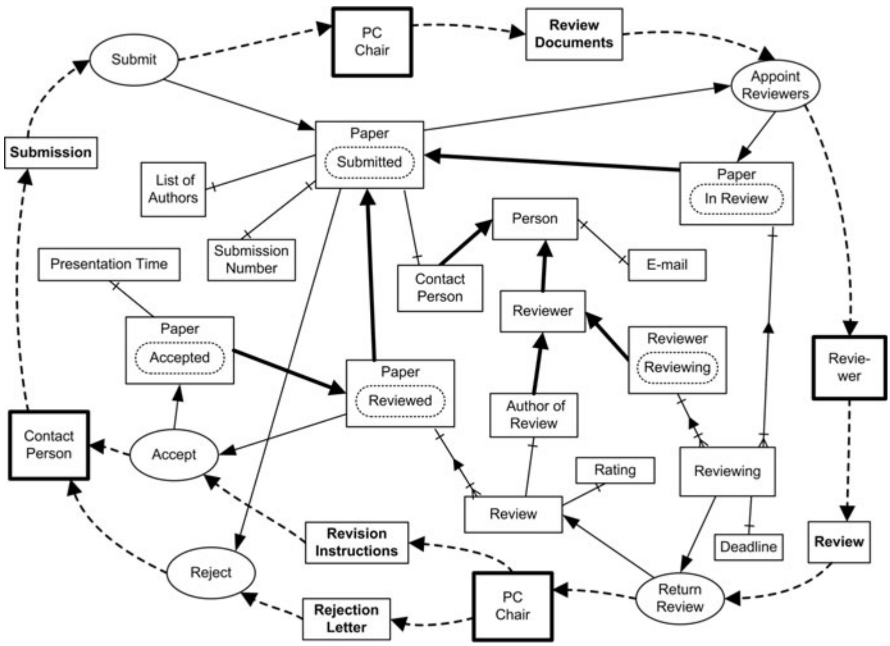


Fig. 8 Interactive, behavioural and structural aspects of a conference management system

from the pre-condition to post-condition classes. The semantic difference must be defined for every transition dependency by using mandatory attribute links. Various types of attribute dependencies in a conference management system are represented in Fig. 8. For example, the Accept action is changing the state of a Paper object from Reviewed to Accepted. Note that each Paper[Accepted] must be characterized by five properties (Presentation Time, Review, List of Authors, Submission number, Contact Person) and Paper[Reviewed] by four properties. The noteworthy semantic difference is represented by the complementary attribute Presentation Time.

6. Refactoring. It is difficult to get initial diagram without inconsistencies and redundancies from the start. Classes and their attributes must be revisited and their semantics examined several times. Otherwise, the diagrams will grow in size. They will be brittle, difficult to comprehend, extend and maintain. The refactoring step is necessary to keep conceptual models clean from inconsistent attributes as well as minimize diagrams as much as possible. The refactoring process (Fowler 1999) does not alter semantics of specification. Refactoring is an essential characteristic of good engineering, because it makes necessary structural changes in order to make modeling clean and understandable. The diagram, which illustrates the outcome of this step, is presented in Fig. 8.

Inheritance mechanism allows sharing attributes via generalization/specialization relations. So, inheritance hierarchies can be used to reduce the diagram. For instance, Accepted Paper class in this diagram inherits attributes from Paper in state Reviewed.

7. Adding the alternative actions. This step is important, because the modeling process should provide with the possibility to demonstrate available alternatives to the main course of events. Note that the Reject action is an alternative to Accept. Therefore, it is ideally should be added at this step. If the IS analysis expert is interested in the main scenario, then the Reject action can be removed together with its termination effect from the diagram in Fig. 8. One of the advantages of service-oriented modeling method is that it allows simple extension of diagrams. The method is very useful for introducing evolutionary extensions, which can be represented by alternative actions. Let us modify the initial description of a conference management system by introducing the following requirement: *a contact person should have a possibility to* ***withdraw*** *a submitted paper from a conference at any time.* Such new requirement would cause a very simple extension of the diagram, which is represented in Fig. 8. Withdraw is an alternative action, which can be represented by interaction dependency: Withdraw(Contact Person - - -> PC chair). The corresponding transition dependency ($\rightarrow$) from Paper[Submitted] to Withdraw action is necessary to specify the termination effects of a Paper object.

Related triggering effects of withdraw action can be recognized from the presented conceptualization. There are four sets of effects, which must be taken into account when analyzing the states of a Paper (see Submitted, In Review, Reviewed and Accepted). Representing the semantics of such behavioral effects in the state-transition, activity and sequence diagrams, would require their significant extensions. The new state-transition diagram should contain four new transitions with their associated events and effects. Four variations of withdraw actions must be added to activity diagram together with the corresponding decision symbols for evaluation of the withdrawal conditions after execution of *Submit*, *Appoint Reviewers*, *Return Review* and *Accept* activities. As a consequence, the complexity of activity diagram would increase dramatically. Additionally, four new sequence diagrams must be introduced for the definition of related effects, which are represented by the diagram in Fig. 8.

5 Concluding Remarks

One of the main goals of this paper was to define and to explain service-oriented modeling process, which takes into account the pragmatic and semantic dimensions of IS conceptualizations. The advantage of pragmatic-driven analysis is that any business process fragment can be defined in terms of service interactions. Any IS specification fragment from the pragmatic point of view can be analyzed as a problem, goal or opportunity. Refinement of service interactions during the modeling process supports semantic integration of IS conceptualizations. Such integrated and incremental modeling method, by adding new semantic details at every step, provides a systematic way to manage the complexity of diagrams. It resembles a way the systems are built from subsystems. Service-oriented method helps to maintain the semantic integrity of static and dynamic aspects in a single holistic representation. Service-oriented diagrams follow the basic conceptualization principle in representing only computationally neutral aspects, which are not influenced by any implementation details. Computation-neutral representations are easier to comprehend for business experts without a technical background.

Pragmatic descriptions aim to provide motivation for change management of enterprise components, which are defined in terms of interactions between service requesters and service providers. Goals, problems and opportunities can be represented by a set of pragmatic dependencies, which are supposed to drive the overall IS design process. Modeling of interaction dependencies supports the principle of separation of concerns, which is one of the main principles for managing complexity of enterprise system specifications. Goal decomposition hierarchy is important as it describes how various high-level objectives are going to be achieved. Goal analysis helps to motivate new enterprise subsystems and to provide a basis for reasoning about semantic completeness of specifications. Goal hierarchies can be also used to specify the desirable changes that are represented in terms of service interactions between different organizational or technical components. The main advantage of pragmatic-driven approach is that it enables bridging the pragmatic and semantic specifications.

Service-oriented method helps in diagnosing the potential problems and managing evolutionary changes in organizations. It provides possibility to introduce new problematic or desired situations, which motivate the necessity of changes. Modeling process helps system developers to visualize, specify and analyze IS requirements. The presented modeling process helps to validate whether the system specifications fulfills its intended purpose. Validation process usually refers back to the user's goals and is expressed by the question 'Are you building the right system?' The proposed modeling method is based on the principles of service-oriented analysis. It facilitates reasoning about the interplay of semantic and pragmatic dimensions across organizational and technical system boundaries. Using a single and semantically integrated model helps to detect the discontinuity of IS specifications. Service-oriented method can be characterized by an improved learning capacity, because the modeling language is able to reduce system architecture evolution complexity.

References

Alter S (2008) Service system fundamentals: work system, value chain, and life cycle. IBM Syst J 47(1):71–85

Chesbrough H, Spohrer J (2006) A research manifesto for service science. Commun ACM 49(7):35–40

Erl T (2005) Service-oriented architecture: concepts, technology, and design. Pearson, Upper Saddle River

Evermann J, Wand Y (2009) Ontology based object-oriented domain modeling: representing behavior. J Database Manage 20(1):48–77

Ferrario R, Guarino N (2008) Towards an ontological foundation for service science. In: Future internet-FIS2008: the first internet symposium, FIS 2008 Vienna, Austria. Revised Selected Papers, Springer, Berlin, pp 152–169

Fowler M (1999) Refactoring: improving the design of existing code. Addison-Wesley, Menlo Park

Glinz M (2000) Problems and deficiencies of UML as a requirements specification language. In: Proceedings of the 10th international workshop on software specification and design, San Diego, pp 11–22

Gordijn J, Akkermans H, van Vliet H (2000) Business process modelling is not process modelling, conceptual modelling for e-business and the web, LNCS 1921. Springer, Berlin, pp 40–51

Gustas R (2010) A look behind conceptual modeling constructs in information system analysis and design. Int J Inf Syst Model Des 1(1):79–108

Gustas R, Gustiene P (2008) Pragmatic-driven approach for service-oriented analysis and design. In: Information systems engineering—from data analysis to process networks. IGI Global, New York

Gustas R, Gustiene P (2009) Service-oriented foundation and analysis patterns for conceptual modelling of information systems. In: Information system development: challenges in practice, theory and education, vol 1. Springer, New York, pp 249–265

Jacobson I, Ng P-W (2005) Aspect-oriented software development with use cases. Pearson, Upper Saddle River

Lee G, Xia W (2010) Toward agility: an integrated analysis of quantitative and qualitative field data on software development agility. MIS Q 34(1):87–114

OMG (2010) Unified modeling language superstructure, version 2.2. www.omg.org/spec/UML/2.2/. Retrieved 19 Jan 2010

Rouse WB, Baba ML (2006) Enterprise transformation. Commun ACM 49(7):67–71

Zachman JA (1987) A framework for information systems architecture. IBM Syst J 26(3): 276–292

Automatic Tool Support for Cardinality-Based Feature Modeling with Model Constraints for Information Systems Development

Abel Gómez and Isidro Ramos

Abstract Feature Modeling is a technique that uses diagrams to characterize the variability of software product lines. The arrival of metamodeling frameworks in the Model-Driven Engineering field (MDE) has provided the necessary background to exploit these diagrams (called feature models) in information systems development processes. However, these frameworks have some limitations when they must deal with software artifacts at several abstraction layers. This paper presents a prototype that allows the developers to define cardinality-based feature models with complex model constraints. The prototype uses model transformations to build Domain Variability Models (DVM) that can be instantiated. This proposal permits us to take advantage of existing tools to validate model instances and finally to automatically generate code. Moreover, DVMs can play a key role in complex MDE processes automating the use of feature models in software product lines.

1 Introduction

The changing nature of technology and user requirements leads us to need multiple versions of the same or similar software application in short time periods. The Software Product Line (SPL) [1] concept arises with the aim of controlling and minimizing the high costs of developing a family of software products in the previous context. This approach is based on the creation of a design that can be shared among all the members of a family of programs within an application domain. The key aspect of Software Product Lines (SPL) that characterizes this approach with respect to other software reuse techniques is how to describe and manage variability, mainly by means of feature modeling as proposed in [2].

A. Gómez · I. Ramos (✉)
Departamento de Sistemas Informáticos y Computación, Universitat Politècnica de València
Cno. de Vera, s/n. Valencia, Spain.
e-mail: agomez@dsic.upv.es

I. Ramos
e-mail: iramos@dsic.upv.es

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_22, © Springer Science+Business Media, LLC 2011

The use of feature models can be exploited by means of metamodeling standards. In this context, the Model-Driven Architecture [3] proposed by the Object Management Group (OMG) is a widely used standard which arises as a suitable framework for this purpose. In this environment, MDE and the Generative Programming approach [4] provides a suitable basis to support the development of SPLs. Moreover, Generative Programming and SPLs facilitate the development of software products for different platforms and technologies.

In this paper we present a prototype that uses current metamodeling tools to define cardinality-based feature models and their configurations. The selected platform is the Eclipse Modeling Framework (EMF) [5]. Moreover, feature models can be enriched with complex model constraints that can be automatically checked by means of the internal Object Constraint Language (OCL) [6] interpreter. All these features of EMF allows developers to start a Software Product Line.

The remainder of this paper is structured as follows: in Sect. 2 we briefly introduce feature modeling and in Sect. 3 we describe how nowadays standards and tools can be used to exploit them. In Sect. 4 our proposal is presented. Related works are discussed in Sect. 5 and in Sect. 6 we present our conclusions.

2 Feature Models and Cardinality-Based Feature Models at a Glance

Feature models are diagrams which express the commonalities and variabilities among the products of a SPL. These models organize the so-called features (*uservisible aspect or characteristic of the domain*) in a hierarchical structure. The basic relationships between a feature and its children are: *mandatory* relationships (which represent the shared design), *optional* relationships, *OR* groups and *XOR* groups.

Cardinality-based feature modeling [7] integrates several of the different extensions that have been proposed to the original FODA notation [2]. A cardinality-based feature model is also a hierarchy of features, but the main difference with the original FODA proposal is that each feature has associated a *feature cardinality* which specifies how many clones of the feature are allowed in a specific configuration. Cloning features is useful in order to define multiple copies of a part of the system that can be differently configured. Moreover, features can be organized in *feature groups*, which also have a *group cardinality.* This cardinality restricts the minimun and the maximun number of group members that can be selected. Finally, an *attribute type* can be specified for a given feature. Thus, a primitive value for this feature can be defined during configuration which is useful to define *parameterized features.*

In feature models is also quite common to describe constraints between features such as the *implies* and the *excludes* relationships, which are the most common used ones. In classic feature models the semantics of these constraints can be expressed by means of propositional formulas [8], thus, it is possible to reason about the satisfiability of the feature model and its configurations.

A configuration of a feature model can be defined as *a valid set of instances of a feature model.* I.e., the relationship between a feature model and a configuration is

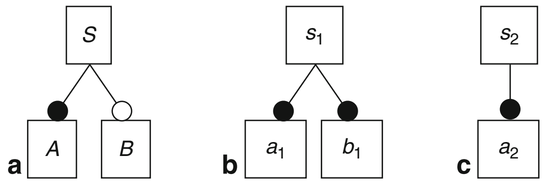


Fig. 1 Example of a feature model (**a**) and the two possible configurations that it represents (**b** and **c**)

comparable to the relationship between a class and an object. In Fig. 1a an example feature model is represented. This feature model represents a system *S*, with two features *A* and *B*. The first one, feature *A*, is mandatory (it must be included in every possible product of the product line), and the second one, feature *B*, is optional (it can be included in a particular product or not). Thus, we have two possible configurations for this feature model, which are represented in Fig. 1b, c.

3 Feature Modeling and Model-Driven Engineering

Model-Driven Engineering (MDE) is a Software Engineering field that over the years has represented software artifacts as models in order to increase productivity, quality and to reduce costs in the software development process. Nowadays, there is increasing interest in this field, as demonstrated by the OMG guidelines that support this trend with the Model-Driven Architecture (MDA) [3] approach. The Meta Object Facility standard (MOF) [9], which provides support for meta-modeling, defines a strict classification of software artifacts in a four-layer architecture (from M3 to M0 layer). As it provides support for modeling and metamodeling, we can use MOF to define cardinality-based feature models by defining its metamodel. Thus, we can define a model which captures the whole variability of the domain, which, in turn, allows us to define any possible configuration of the model. But, what is more, it can also be used to define model-based transformations that enable the use of feature models and their configurations in other complex processes. Figure 2 shows where feature models and configurations fit in the four-layer MOF architecture.

3.1 MDA in Practice: Industrial Tool Support

The Eclipse Modeling Framework (EMF) [5] can be considered as an implementation of the MOF architecture. Ecore, its metamodeling language, can be placed at layer M3 in the four-layer architecture of the MOF standard. By means of Ecore, developers can define their own models which will be placed at the metamodel layer (M2). An example of such metamodels is the one to build cardinality-based feature models. Finally, these Ecore models can be used to automatically generate graphical editors which are capable of building *instance models,* which will be placed at M1 layer. In the case of feature modeling, these *instance models* are the feature models. The left column on Fig. 3 shows this architecture.

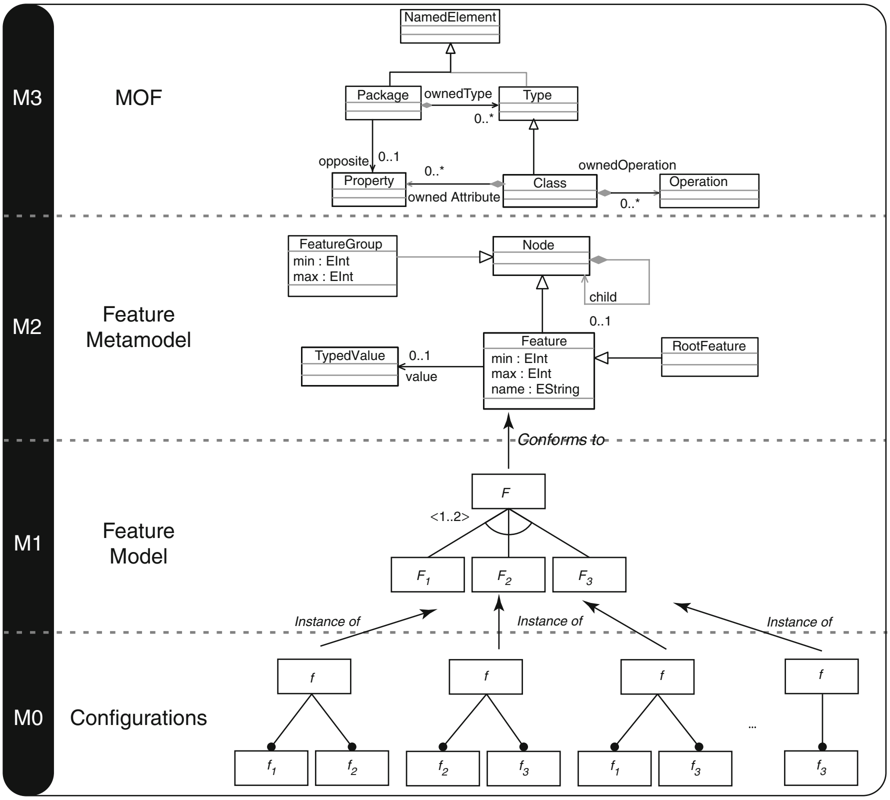


Fig. 2 Definition and configuration of feature models in the context of MOF. The EMOF language is represented in a simplified way in the level *M3*. In the level *M2* the metamodel for cardinality-based feature models is represented by using the MOF language (also in a simplified way). In the level *M1* feature models are described. Some configurations of the example feature model are shown at level *M0*

As can also be seen in Fig. 3, the M0 layer is empty. This is a limitation of most of the modeling frameworks which are available today. As said, EMF provides a modeling language to define new models and their instances, but this framework only covers two layers of the MOF architecture: the metamodel and the model layers. However, in the case of feature modeling we need to work with three layers of the MOF architecture: metamodel (cardinality-based feature metamodel), model (cardinality-based feature models), and instances (configurations).

Figure 3 shows how to overcome this drawback: it is possible to define a model-to-model transformation in order to convert a feature model (i.e. the model represented by *Feature model* which can not be instantiated) to an Ecore model (i.e. the *Domain Variability Model*, DVM, which represents the Feature model as a new class diagram). Thus, it is possible to represent a feature model at the metamodeling layer, making the definition of its instances possible. This way, developers can take advantage of EMF again, and automatically generate editors to define feature model configurations,

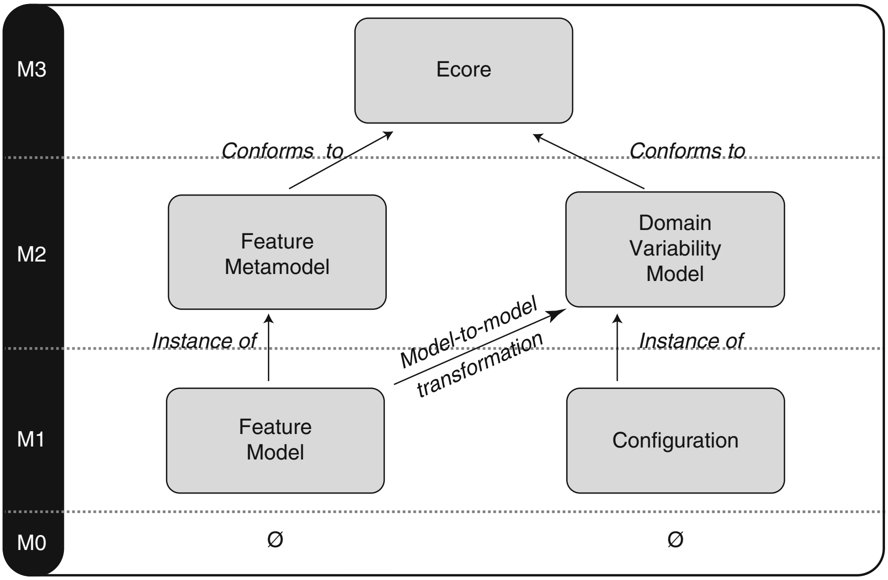


Fig. 3 EMF and the fourlayer architecture of MOF. The Ecore language is placed at the *M3* layer, the Ecore models (such as the metamodel for cardinality-based features models) are placed at the *M2* layer, and model instances (feature models) are placed at the *M1* layer. It is noteworthy that EMF can not represent more than 3 layers

and validate them against their corresponding feature models thanks to their new representation, the DVM. Moreover, as the DVM is an Ecore model (a simplified UML class diagram) we automatically obtain support to check complex constraints (by using OCL) over the feature model configurations.

4 Our Approach

Based on the concepts presented in the previous section and using EMF, we have developed a tool that allows us to automate several steps in order to prepare a feature model that can be exploited to develop a SPL in the context of MDA. In this sense, our tool provides:

- Graphical support to define (a variant of) cardinality-based feature models with model constraints expressed by using a constraint language.
- Support to automatically generate DVMs from feature models that capture all the variability of the application domain (including complex model constraints), allowing the developers to use them in model transformations.
- Support to transform model constraints to OCL expressions.
- Configuration editors, which will assist the developers.
- Capabilities to check the consistency of a configuration against its corresponding feature model by using pre-built OCL engines.

All these tasks are automatically supported by using MDE techniques (modeling, metamodeling, model transformations and code generation). The following subsections describe how this process has been implemented.

4.1 Cardinality-Based Feature Metamodel

The basis of our work is the cardinality-based feature metamodel, which permits to define feature models. Figure 4 shows our feature metamodel. Such metamodel has been defined taking into account that every element will have a different graphical representation. This way, it is possible to automatically generate the graphical editor to draw feature models based on such metamodel.

4.1.1 Feature Models Structure

In Fig. 4, a feature model is represented by means of the *FeatureModel* class, and a feature model can be seen as a set of *Features*, the set of *Relationships* among them and the set of model constraints (*modelConstraints* role) that are applied to it. A feature model must also have a root feature, which is denoted by means of the *rootFeature* role. As can be seen in Fig. 4, our proposal represents explicitly the relationships between features. Thus, it represents in an uniform way the hierarchical relationships (*StructuralRelationship* class) and the restrictions between features (*RestrictionRelationship* class). The classification of these relationships is explained in detail in [10].

4.1.2 Feature Model Constraints

As was pointed out in Sect. 2, it is quite common in feature modeling to have the possibility to define model constraints in order to describe more precisely which configurations should be considered as valid. Typically, these constraints are described by means of implication or exclusion relationships. This kind of relationships are the *binary and horizontal relationships* that our metamodel provides.

The *binary and horizontal relationships* are specified between two features and they can express constraints (coimplications, implications and exclusion) or dependencies (use). The first group applies to the whole set of instances of the involved features, however, the second one allows us to define dependencies at instance level, i.e.:

- *Implication* ($A \rightarrow B$)*:* If an instance of feature A exists, at least an instance of feature B must exist too.
- *Coimplication* ($A \leftrightarrow B$)*:* If an instance of feature A exists, at least an instance of feature B must exist too and vice versa.
- *Exclusion* ($A \;\times\!\!\longrightarrow\!\!\times\; B$)*:* If an instance of feature A exists, can not exist any instance of feature B and vice versa.

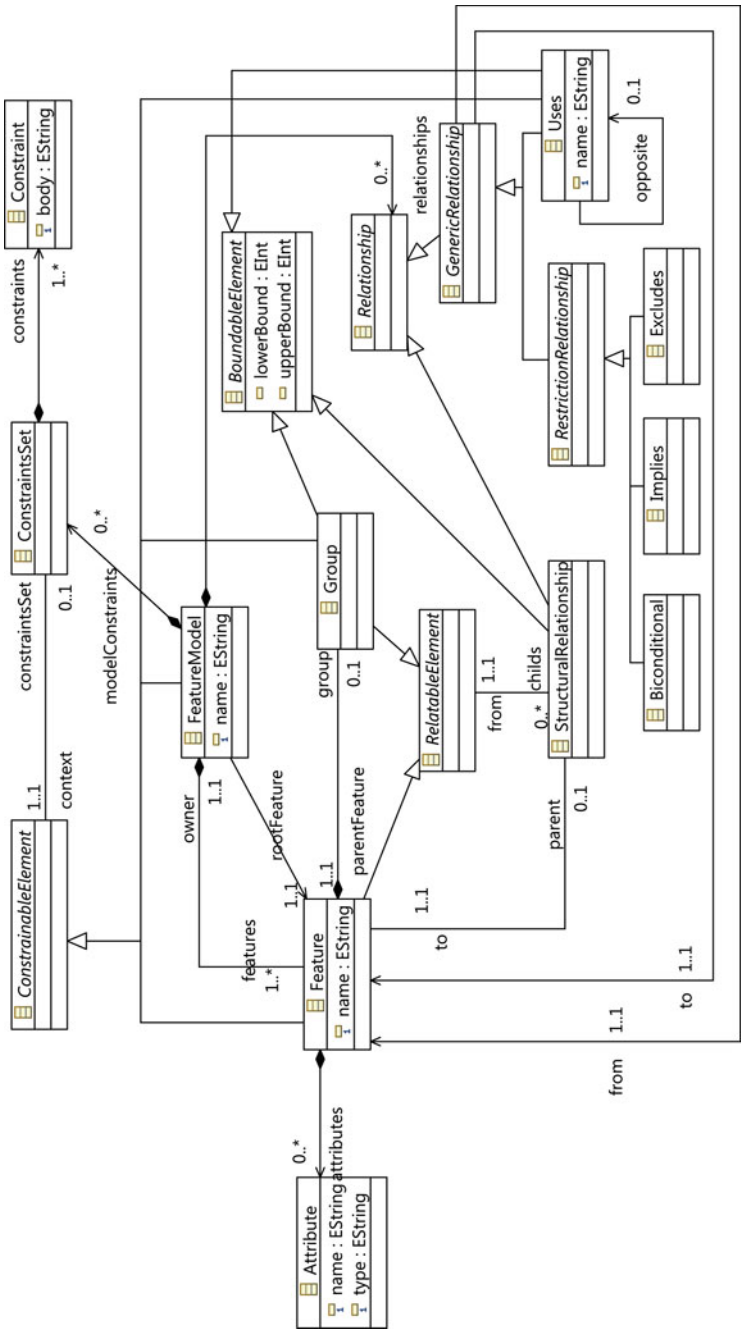


Fig. 4 Cardinality-based features metamodel

- *Use (A - - → B):* This relationship will be defined at configuration level, and it will specify that an specific instance of feature A will be related to one (or more) specific instances of feature B as defined by its upper bound (n).

Besides these kind of relationships that describe coarse-grained restrictions, our metamodel provides capabilities to describe fine-grained restrictions. These restrictions are stored in a *Constraints Set* instance, and can be applied to any subclass of the abstract class *ConstrainableElement* (*context* role), i.e., *FeatureModel*, *Feature*, *Group* or *Uses*. The restrictions are expressed as a textual expression (body attribute of the *Constraint* class).

To describe these fine-grained restrictions we propose a constraint language, called *Feature Modeling Constraint Language* (FMCL). FMCL is a formal language without side-effects (does not modify the model instances) whose syntax is based on the widely known *Object Constraint Language* (OCL) and its semantics are defined by a set of patterns that describe the equivalences between FMCL expressions and OCL expressions.

4.1.3 Cardinality-Based Feature Modeling Editor

Following the Model-Driven Software Development (MDSD) approach, graphical editors can be automatically generated from the metamodel presented: i.e., the cardinality-based feature modeling editor. This editor allows us to easily define new feature models. To obtain this graphical editor, the Graphical Modeling Framework (GMF) [11] has been used.

Figure 5 shows what this editor looks like. The palette is located on the right side of the figure, and shows the tools that can be used to define the feature models. In the

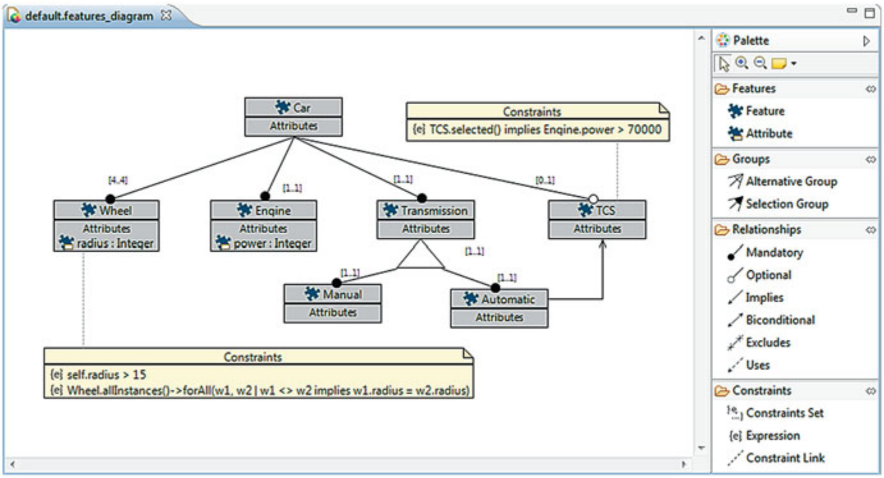


Fig. 5 Example feature model

canvas an example feature model is shown. This feature model describes a simple product line for cars. A car must have four wheels (of a given radius), one engine (of a specific power in watts) and a transmission (which can be manual or automatic). As an optional equipment the car can have a Traction Control System (TCS). The feature model also describes four constraints: the arrow between the feature TCS and Automatic states that if an automatic transmission is selected, the TCS must be selected too; the annotation attached to the TCS feature states that the TCS can only be selected if the power of the engine is higher than 70,000 watts; and finally, the annotation attached to the Wheel feature specifies that the radius of the instances of the wheel must be higher than 15 inches and that all the wheels must be of the same size.

4.2 *The Domain Variability Model*

The Domain Varibility Model is a class diagram (an Ecore model) whose instances are equivalent to the configurations of a feature model. It is intended to ease the definition of feature model configurations in EMF as was explained in Sect. 3.1. This model can be automatically generated by means of a model-to-model transformation. Following the MDA guidelines, this transformation is defined by using the Relations language defined in the QVT standard [12]. In order to integrate and execute this transformation process in our prototype, a custom tool based on the mediniQVT [13] transformations engine has been built.

4.2.1 The Structure of the DVM

As can be observed in Figs. 5 and 6, the transformation regarding to the structure of the DVM is almost a one-to-one mapping. For each *Feature* of the source model an *EClass* (with the same *name*) is created. All the classes are created inside the

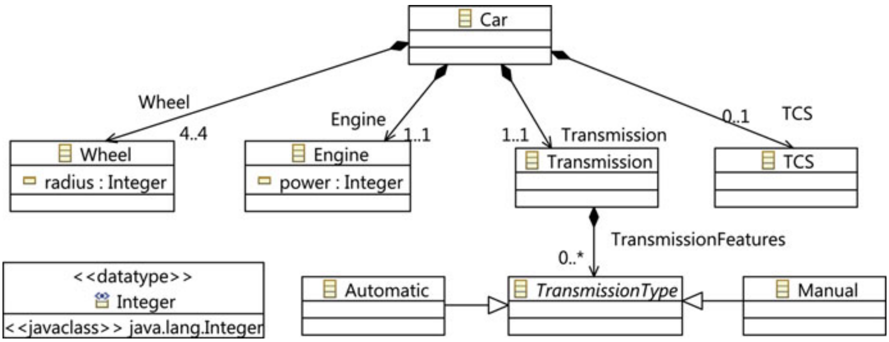


Fig. 6 Generated DVM for the example feature model. In the figure, model constraints (OCL annotations) have been omitted for clarity purposes

same *EPackage*, whose name and identifier derives from the feature model name. Moreover, for each feature *Attribute*, an *EAttribute* in its corresponding *EClass* is created in the target model. Any needed *EDataType* is also created.

Regarding to the relationships, for each *StructuralRelationship* from a parent *Feature*, a containment *EReference* will be created from the corresponding *Eclass* and for each *Group* contained in a *Feature* a *containment EReference* will be created from the corresponding *EClass*. This *EReference* will point to a new abstract class, whose name will be composed by the *Feature name* and the suffix "Type". Additionally, an *EClass* will be generated for each *Feature* belonging to a *Group*. Moreover, each one of these *EClasses* inherit from the abstract *EClass* that has been previously created. Finally, for each *Uses* relationship between two *Features*, an *EReference* will be created in the target model. This *EReference* will relate two *EClasses* whose names will match the *Features* names.

4.2.2 Constraints Over the DVM

The *restriction relationships* and *model constraints* (FMCL expressions) are mapped to OCL expressions in the DVM. The mappings to transform the *restriction relationships* to OCL expressions is described in [10] in detail.

The FMCL expressions are mapped to OCL expressions taking into account the mappings explained in Sect. 4.2.1. Figure 5 shows an example of this. As can be seen on the constraint that applies to the Wheel feature, a FMCL expression can be expressed directly using the OCL syntax. This way, an FMCL expression is directly transformed to an OCL invariant. The context of the invariant corresponds to the name of the *ConstrainableElement* that is linked to the constraint (dashed line in the figure) and the text of the expression remains the same:

```
context Wheel
inv: self.radius > 15
inv: Wheel.allInstances()->forAll(w1, w2 |
                                  w1 <> w2 implies w1.radius = w2.radius)
```

However, although the FMCL expressions are almost the same than an OCL invariant, some simple conventions have been adopted to make the definition of model constraints closer to the feature modeling context. The semantics of these additions are defined by means of transformation patterns (see Table 1).

Table 1 Summary of transformation patterns (FMCL to OCL)

FMCL expression pattern	Equivalent OCL definition
ConstrainableElement	*ConstrainableElement*.`allInstances()`
ConstrainableElement.property op expression	*ConstrainableElement*.`allInstances()` `->forAll`*(property op expression)*
ConstrainableElement.`selected()`	*ConstrainableElement*.`allInstances()` `->notEmpty()`
FeatureName.`childs()`	*FeatureName*`Type.allInstances()`

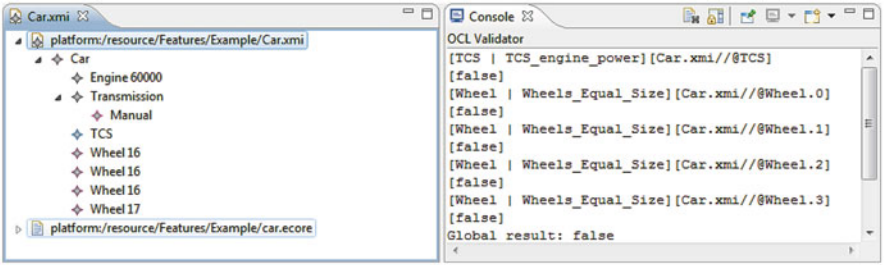


Fig. 7 Example of an unsuccessful configuration check

An example of the application of some of these patterns can be seen in the constraint attached to the TCS feature (Fig. 5). The example constraint is transformed to the following OCL expression:

```
context TCS inv:
TCS.allInstances()->notEmpty() implies Engine.allInstances()->forAll(power >
    70000)
```

4.3 *Creating and Validating Configurations*

In order to create new configurations of feature models it is not necessary to use any custom tool. As far as we have a DVM which captures the same variability than the original feature model, developers can use the standard Ecore tools. However, in EMF there is not a default tool to check OCL invariants which are directly stored as *EAnnotations* in Ecore models themselves. Thus, we have built an extension which can take advantage of the OCL invariants that have been automatically created in the previous transformation step.

Figure 7 shows an example configuration. It shows a car configuration with manual transmission, TCS, 4 wheels and engine. The radius of three of the wheels is 16 inches, and the radius of the fourth is 17 inches. The power of the engine is 60,000 watts. This configuration is invalid conforming to the restrictions applied to the metamodel. When the configuration is invalid the checking process is unsuccessful. In this situation, the prototype console shows a summary with the constraints that are not met, and which are the problematic elements as the figure shows.

5 Related Works

Feature modeling has been an important discussion topic in the SPL community, and a great amount of proposals for variability management have arisen. Most of them are based in the original FODA notation and propose several extensions to it

[14]. Our work is closely related with previous research, however, there are several distinctive aspects:

Our work describes a prototype to define and validate configurations of feature models. Previous work has been also done in this area, such as the *Feature Modeling Plugin* [15]. The main difference with our work is that configurations are defined in terms of the feature metamodel and both models and configurations coexist at the same layer. Thus, in order to be able to deal both with models and configurations it is necessary to build complex editors (as they must guarantee that the specialization process is properly done), and whats more, those artifacts can not be easily used in complex MDE processes.

Some previous works have already represented feature models as class diagrams [7, 16]. However, in [7] no set of transformation rules define the mappings between features and classes. In this work, OCL is also presented as a suitable approach to define model constraints, but there is no automatic generation of OCL invariants as the transformation is not clearly defined. In turn, [16] presents a set of QVT rules to define the mappings. However, in this case, neither model constraints nor configuration definitions support is presented.

In [8] a proposal for feature constraints definition and checking is done, representing features as propositions and restrictions among them as propositional formulas. However, this approach is not suitable when features can have typed attributes which can not be expressed by this kind of formulas. We state that more expressive languages are needed, such as FMCL/OCL.

6 Conclusions

In this paper we have presented a framework[1] to define and use feature models in a MDE process. This framework addresses two issues: first, the inability of current metamodeling tools to deal simultaneously with artifacts located in all the MOF layers; and second, the complexity to define model constraints in feature models where features can be cloned and can have attributes. These problems have been solved by transforming feature models to Domain Variability Models that can be instantiated and reused in future steps of the MDE process.

Our tool has been designed following the MDE principles and a metamodel for cardinality-based feature modeling has been defined. By means of generative programming techniques, a graphical editor for feature models has been built. Feature models defined with this editor are automatically transformed to DVMs that are used to define configurations of feature models. Although several tools to define feature models and configurations in the last years have arisen, our approach has several advantages against previous approaches: (1) the infrastructure that we propose to build configurations is simpler and more maintainable, as it is built following the MDSD

[1] This framework is supported by a prototype that can be downloaded from http://issi.dsic.upv.es/∼agomez/feature-modeling.

guides; (2) configurations are actually instances of a feature model (expressed by means of the DVM), so we can take advantage of the standard EMF tools; (3) as feature models are described by DVMs that can be instantiated, both models and configurations can be used in other MDE tasks; (4) having a clear separation between feature models and configuration eases the validation tasks as they can be performed by means of built-in languages; and (5) as the transformation between feature models and DVMs is performed automatically by means of a declarative language we can trace errors back from DVMs to feature models.

It is noteworthy to remark the importance of using feature models and configurations at different layers. In [17] an example where this architecture is used to integrate feature models in a MDE process is shown. This work describes how a model transformation with multiple inputs (feature models and functional models) is used to generate a software architecture automatically.

Acknowledgements This work has been supported by the Spanish Government under the National Program for Research, Development and Innovation MULTIPLE TIN2009-13838 and the FPU fellowship program, ref. AP2006-00690.

References

1. Clements P, Northrop L, Northrop LM (2001) Software Product Lines: practices and patterns. Addison-Wesley, Boston
2. Kang K, Cohen S, Hess J, Nowak W, Peterson S (1990) Feature-oriented domain analysis (FODA) feasibility study
3. Object Management Group (2003) MDA guide version 1.0.1. http://www.omg.org/docs/omg/03-06-01.pdf
4. Czarnecki K, Eisenecker UW (2000) Generative programming: methods, tools, and applications. Addison-Wesley, New York
5. EMF. http://download.eclipse.org/tools/emf/scripts/home.php
6. Object Management Group (2006) OCL 2.0 Specification. http://www.omg.org/cgi-bin/doc?formal/2006-05-01
7. Czarnecki K, Kim CH (2005) Cardinality-based feature modeling and constraints: a progress report
8. Batory D (2005) Feature models, grammars, and propositional formulas. Springer, New York, pp 7–20
9. Object Management Group (2006) Meta object facility (MOF) 2.0 core specification (ptc/06-0101). http://www.omg.org/cgi-bin/doc?formal/2006-01-01
10. Gómez A, Ramos I (2010) Cardinality-based feature modeling and model-driven engineering: fitting them together. In: Fourth international workshop on variability modelling of softwareintensive systems (VAMOS'10). Linz, Austria
11. Eclipse Organization (2006) The graphical modeling framework. http://www.eclipse.org/gmf/
12. Object Management Group (2005) MOF QVT final adopted specification
13. ikv++ technologies AG, "ikv++ mediniQVT website." http://projects.ikv.de/qvt
14. Chen L, Babar MA, Ali N (2009) Variability management in software product lines: a systematic review. In: Proceedings of the 13th international software product lines conference (SPLC'09), San Francisco, USA
15. Antkiewicz M, Czarnecki K (2004) FeaturePlugin: feature modeling plug-in for Eclipse. 2004 OOPSLA workshop on eclipse technology eXchange, pp 67–72

16. Laguna MA, Gonzälez-Baixauli B, Marqués Corral JM (2008) Feature patterns and multiparadigm variability models. Tech. Rep. 2008/2001, Grupo GIRO, Departamento de Informätica, May 2008
17. Cabello ME, Ramos I, Gómez A, Limón R (2009) Baseline-Oriented Modeling: an MDA approach based on software product lines for the expert systems development. Asian conference on Intelligent information and database systems, vol 0, pp 208–213

Towards a Conceptual Model for Trustworthy Skills Profiles in Online Social Networks

Till Haselmann, Axel Winkelmann and Gottfried Vossen

Abstract For many users online profiles displaying other people's skills are increasingly important, e.g., when contracting freelancers or finding candidates for a job opening. However, current profiles found in information systems offer either unstructured free text that is hard to handle efficiently or simplistic rating schemes that do not convey meaningful information. In addition, it is unclear how trustworthy the information on the profile is. In this paper, we propose a novel approach to online skills profiles based on users' confirmations and the SkillRank credibility ranking and describe its prototype implementation. As spadework, we set forth six basic requirements for online skill evaluations which should generally be reflected in corresponding IS design.

1 Introduction

In the light of the growing dissemination of online social networks, the significance of online profiles is increasing for a wide variety of domains. Both in leisure networks, such as Facebook or the German StudiVZ, and in business networks, such as Plaxo, LinkedIn or XING, profile pages are a pivotal means for judging a person. Particularly in the context of serious business contacts, it can be of monetary value to know how trustworthy the information on a profile page is, e.g., if a headhunter needs to assess whether the person in question truly fits a specific job description or whether it is just pretense.

Having been engaged massively in social activities on the Internet since the rise of the Web 2.0 phenomenon [9, 13], people have become interested in not only contemplating other people's virtual profiles but also deducing their real-world skills from

T. Haselmann · A. Winkelmann · G. Vossen (✉)
European Research Center for Information Systems (ERCIS), WWU Münster, Münster, Germany
e-mail: haselmann@ercis.uni-muenster.de

A. Winkelmann
e-mail: winkelmann@ercis.uni-muenster.de

G. Vossen
e-mail: vossen@ercis.uni-muenster.de

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_23,

these profiles or recommending them to possible employers. One crucial aspect of a meaningful skills profile is the rating of individual skills. Only by means of a rating mechanism—however simple or elaborate—it is possible to provide a differentiated skills profile that also (at least rudimentarily) reflects strengths and weaknesses of the person in question.

Traditionally, an important offline "system" for evaluation and rating of various entities, such as products and people, has always been and still is word of mouth, i.e., gossip [7, 5]. All social networks nowadays provide functionality for communication and interaction in various ways [3] and most also feature functions for evaluating objects, profiles, or real-world skills.

The advantages of online—in contrast to offline—evaluations are scalability and formalization [1, 8]. Scalability in this context means that estimates can be gathered from and communicated to a multitude of parties, independent of time and place [18]. Users can access a huge number of evaluations provided by other users in an easy and cost-efficient manner [5]. According to [6], especially the fast diffusion of up-to-date information is a major advantage. In addition, evaluations become more comprehensible through unification of gathering, aggregation and presentation and hence their acceptance increases [17]. The formalization of the results also offers new possibilities for automated processing, e.g., enhanced search, structured comparisons or job matching.

However, the evaluation or rating functionality provided is still very limited and mostly constrained to direct rating schemes (cf. [21]). In online business networks, such as XING, Viadeo or LinkedIn, users can advertise their alleged hard and soft skills on their profile pages using free text, enforcing neither structure nor truthfulness and significantly reducing the utility of these claims. On the other hand, there are many formalized rating algorithms on the Internet, but only a few allow rating user skills. For instance, Web pages such as RateMDs.com for the rating of medical services or RateMyTeacher.com for the rating of high school employees allow a simple assessment of professional skills. Generally, these mechanisms are simplistic and their results not very meaningful [21].

Acknowledging that people will always want to present themselves in a favorable light, especially when it comes to the job market [10], we strive for a mechanism that allows for positive profiles—leaving out a person's negative aspects. On the other hand, we already argued that reliable profiles are desirable. So the approach needs to include a mechanism which prevents false statements to appear in the profiles while reinforcing true claims. Coincidentally, the evaluation of people has to be much more meticulous than that of innate objects because the results may have severe negative effects on that person's life. Measuring skills and assessing people in general is, however, a complex endeavor whose validity is highly dependent on a sound theoretical approach [19]. Considering these aspects, it follows that providing IS developers with a set of guidelines for integrating assessment functionality into their IS based on a solid theoretical foundation is a favorable goal.

In that light, our contributions are these: We first state six basic requirements for online skill evaluations and, consequently, propose a novel approach for the unsupervised creation of skills profiles in a social network that addresses these requirements.

The approach provides a structured presentation of a user's skills. The credibility of the claims is provided by the users' confirmation in their personal networks and manifested as a ranking based on the SkillRank algorithm described as well. Finally, we describe a reference implementation that was developed in cooperation with a large European social network.

The remainder of this paper is organized as follows: After a short theoretical background in Sect. 2, we set forth the requirements for online skills evaluation in Sect. 3. Addressing them, we present the new approach in Sect. 4. After that, we sketch the prototype implementation in Sect. 5. Section 6 exhibits some limitations of our approach as well as a brief roadmap for future work. Section 7 concludes the paper.

2 Theoretical Background on Skill Evaluations on the Internet

The basic idea of online evaluation systems is to let users evaluate entities by means of web applications and hence to collect, aggregate, and distribute estimates [18]. The aggregated estimates about an entity, a person, or his or her skills can be used to derive a score, e.g., a trust or reputation score, which can then be communicated to other parties. The scores can assist these parties in deciding whether or not to transact with certain other parties in the future [12].

The use of online evaluation systems requires an adequate design of the underlying mechanisms. According to [4], one of the main decisions refers to the gathering of information. Operators of such systems must determine which users are allowed to rate which entities. Especially, the evaluator's capability of evaluating an entity and his relationship to the evaluated entity needs to be considered. Furthermore, deliberate manipulations by single users must be avoided [8].

In a state-of-the-art study of 102 rating mechanisms [21], it is concluded that evaluation mechanisms are kept very simple in general. Various entities such as people, skills, products or services are evaluated by simple ratings, mostly based on scales. In few cases there are relative evaluations (evaluation of characteristics of one entity compared to those of another one). However, according to [21] there is hardly any suitable mechanism for competency evaluation on the Internet.

The increasing relevance and spread of rating systems forms a new distinct research field. In this context, socio-technical systems as used in the Web 2.0 context may offer new opportunities [20]. According to [16], there is an increasing need to do research on the forms, effects and validity of rating systems. Nevertheless, the recent analysis [21] did not identify any mechanisms that explicitly address the possibilities of evaluating or presenting competencies with the help of social graphs and hence of relationships between various users on the Internet. Thus, with our approach we contribute to the body of knowledge regarding the design of trustworthy online skills profiles in social networks.

Motivated by [2], we have investigated the similarities of website link structures and social structures and have started our research with a closer look at the Page Rank algorithm [15]. In essence, SkillRank is an adaptation of PageRank to the context considered here.

3 Six Requirements for Online Skills Evaluation

In order to formalize the discussion on "reliable profiles" up to this point, we now deduce six requirements that can be used to judge whether the approach in question indeed allows reliable profiles that convey substantial information about a person while being apt for large-scale social networks. These requirements are mainly derived from the literature. A schematic overview of them is shown in Fig. 1. Together, Requirements 1–4 ensure that the approach delivers reliable, i.e., "objective", results suitable for serious business networks. Note that these requirements—specifically 1 and 2—generally rule out direct rating schemes. 5 and 6 ensure that the approach is feasible even in larger social networks where there is no moderator that can watch over the generated evaluations.

On a sidenote, in order for the user to accept and use the application, he or she must always remain in full control of what appears on the profile. As this is an implicit requirement for any profile page on the Internet, we do not state it explicitly as part of our requirements for skills evaluation.

1. **Impede Deliberate Manipulations** As the skills profiles are aimed to be used in sensitive areas such as the job market, the evaluation mechanism needs to impede deliberate manipulations (cf. [8]). This is quite easy to see: Positive evaluations of one's own skills may help in applying for a new job, establishing trust in e-commerce transactions, etc. Hence, it is most likely that some people will try to deliberately manipulate their own or a third party's skill evaluations in a palliating or decrying way. For example, they may only ask friends for their very positive evaluations or a group of students may arrange that they collectively give a bad rating to a teacher. This problem may even be extended to groups of people cartelizing in order assign very good ratings within the group and bad ratings to outsiders. An online evaluation must ensure that such fraudulent behavior is prevented as best as possible.
2. **Stifle Affronts and Revenge Evaluations** Many events internal or external to the social network, such as a personal quarrel, may cause people to feel annoyed with one another. In such situations, it is very likely that *revenge evaluations*

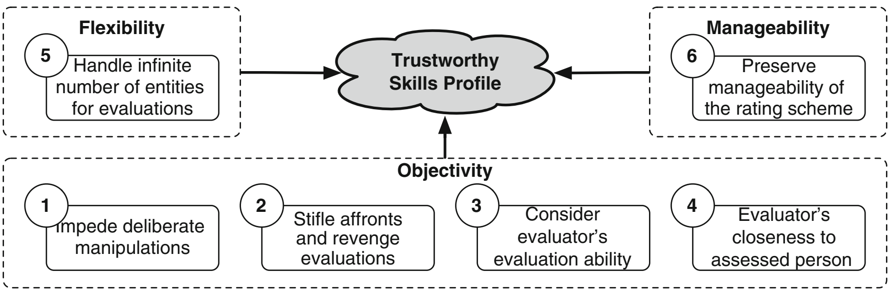


Fig. 1 Requirements for online skills profiles

appear on the platform, i.e., users giving deliberately and unjustly bad ratings, as observed by [18]. This is especially true for direct rating schemes. Apart from the rating, a revenge evaluation may contain insulting comments that appear on the user's profile (cf. [18]). Thus, an online evaluation must ensure that no offending content is published on the users profile without the user's consent. In addition, the revenge ratings should not or at least not significantly influence the evaluation score. Note that any regular direct rating scheme as used on the web today cannot completely fulfill this requirement without relying on moderators (cf. req. 6).

3. **Consider Evaluator's Evaluation Ability** Only people with suitable knowhow will be able to properly evaluate other people's skills, e.g., a person who is not able to speak English will not be able to properly judge another person's proficiency in that language. However, he might be able to judge whether someone is speaking English at all (without being able to note mistakes or bad articulation). An online evaluation needs to consider this in order to produce more reliable ratings (cf. req. [21]).
4. **Consider Evaluator's Closeness to Assessed Person** The closeness of two people has an influence on their mutual evaluations [4]. On the one hand, close relationships between evaluator and evaluated person may help in precisely assessing skills. On the other hand, however, close relations may lead to unjustified evaluations in order to win favor or because of inclination or dislike towards that person. While it is certainly not decidable without detailed data whether and in which way the relationship between two people influences the evaluation, an online evaluation should strive to consider this aspect, although the only feasible option may be to exclude it deliberately.
5. **Handle an infinite number of entities for evaluations** It is generally impossible to identify a common set of suitable skills (especially hard skills) for all users beforehand. Furthermore, each person has different skills. Hence, online skill evaluation needs a high degree of individual skill selection freedom. In consequence, evaluation results for individuals may hardly be comparable. For example, one person may be interested in dancing in general and asks for evaluations on his dancing skills. Another one may be interested in Latin dancing only. An online evaluation must be able to handle an infinite number of skills or provide, if possible at all, an exhaustive taxonomy of skills a priori.
6. **Preserve manageability of the rating scheme** In popular social networks, the number of participants quickly reaches a level where the social network provider does not have the resources to supervise the rating process and check for fraud or offensive behavior. Thus, the evaluation approach must provide reliable results independently of a moderator that intervenes and filters out offensive comments or evaluations. This is a hard requirement for any online evaluation that addresses a generally unbounded number of participants and is not confined to clearly limited, very small groups.

4 Conceptual Model

In order to meet the requirements from Sect. 3, we propose a new approach that relies on *confirmations* rather than direct ratings. Using confirmations, the task of evaluating is basically delegated to the user himself. Each user is able to create a skills profile that he or she finds representative for his or her competencies and that contains only those aspects he or she wants to publish. The alleged skills are then substantiated by *experiences* that the user provides together with specific experience levels. The experiences represent verifiable facts about the user employing or showing the skill at a particular proficiency level. By giving a short textual description, the user can convey the situation in his words. For example, a programmer may want to advertise his Java expertise that he has gained (among other occasions) from a recent programming project. For that, he creates a skill "Java SE programming" and provides an experience describing his programming project. Associated with the experience, he specifies the experience level "expert" because the project involved a lot of very tricky Java programming. Colleagues or teammates that have witnessed the particular skill demonstration can then *confirm* the experience. The confirmation includes a direct reference to the name and the profile page of the confirming user to allow easy verification of his reputation. A textual note can be attached to the confirmation, e.g., describing the evaluator's confidence about the confirmed experience and its level; the confirmed level, however, cannot be modified.

Given the skills profile, a user's actual skill level can be derived from the confirmed rated experiences by calculating a weighted average over the confirmed experience levels. By implication, the user is not able to set a definite skill level himself but has to "suggest" a skill level for each experience. If other people find this suggestion appropriate, they can confirm it. The confirmation is backed by their names and reputations. Without loss of generality, we constrain our considerations in this paper to the three experience levels "*nov*ice", "*adv*anced" and "*exp*ert" with associated weights $\mathrm{w}_{\mathrm{nov}} = 1$, $\mathrm{w}_{\mathrm{adv}} = 2$ and $\mathrm{w}_{\mathrm{exp}} = 3$. This simplified view can easily be extended to allow for a more detailed assessment in a straightforward manner.

Let $\mathrm{C} = C_{\mathrm{nov}} \cup C_{\mathrm{adv}} \cup C_{\mathrm{exp}}$ be the sets of confirmed experiences at the respective experience levels. A user's actual skill level l is then defined as

$$l = \frac{w_{\mathrm{nov}} \cdot |C_{\mathrm{nov}}| + w_{\mathrm{adv}} \cdot |C_{\mathrm{adv}}| + w_{\mathrm{exp}} \cdot |C_{\mathrm{exp}}|}{|C|}.$$

For w_i as defined above, this results in an average skill level $1 \leq l \leq 3$ for each skill which can be interpreted as follows. If $l = 1$, the user is an absolute novice. A level $l = 2$ indicates an advanced user, and $l = 3$ states that the user is an expert. Intermediate values are likely to occur: For example, the aforementioned Java programmer may have a level of $l = 2.4$ for his "Java SE programming" skill indicating him as an advanced user with some expert knowledge, almost half way to being an expert.

While this approach produces a rating for the user's skill level with an implicit credibility provided by the confirming users, the credibility is still not obvious. In order to make the credibility of the claim explicit, we adopt a Markov chain approach

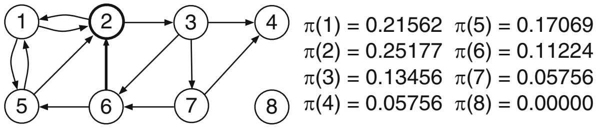


Fig. 2 Sample skill graph with SkillRank values for $\alpha = 0.95$

as proposed in [2], which has also been adopted in a similar manner for Google's PageRank algorithm [15] that measures the importance of Web pages. For the Web, the intuitive idea of importance can be formulated as a recursive conjecture: *An important Web page is one that is linked to by other important Web pages* [14]. Despite having been extended and tweaked in various ways, the PageRank still remains a very robust measure for page importance.

Returning to skills profiles, we can formulate a similar conjecture for our skill evaluation mechanism: *A person's claim to possess a certain skill at a specific level is credible if other people who are credible for the same skill confirm it.*

This conjecture leads eventually to a directed graph $G = (V, E)$ for each skill whose nodes $v_{\mathrm{i}} \in V$ represent the users having that skill. The edges $(v_i, v_j) \in E$ of the graph represent user i's confirmation of some piece of user j's experience. We refer to the resulting graph as the *skill graph* for a specific skill. Accordingly, the algorithm to calculate the credibility ranking is called SkillRank, in analogy to the PageRank.

The example in Fig. 2 shows a very small skill graph with eight users. Supposing node 2 represents the Java progammer, the edge (v_6, v_2) in the graph indicates that user 6 has confirmed some part of user 2's Java experience. The most credible users for this skill are user 2 and 1, user 2 having the highest credibility rank of $\pi(2) \approx 0.25177$. Note that the $\pi(2)$ is not the same as the user's skill level.

In order to ensure the existence of a unique solution to the fixed-point problem, we need to make three adjustments to the graph—or rather the corresponding transition matrix—somewhat similar to Google's adjustments as described in [14]. The first adjustment is that all dangling nodes are treated as though they were connected to all nodes in the graph. This is necessary to eliminate "rank sinks", i.e., nodes that would accumulate an unjustifiably high credibility rank [14]. Figuratively speaking, the accumulating credibility flow is skimmed and re-inserted into the calculation in the fairest manner possible.

The second adjustment is to apply a dampening factor $0 \ll \alpha \leq 1$ so that the cycles in the graph are broken up and the convergence to a unique solution is ensured. The dampening factor introduces a small random element into the calculation so that a portion of $(1 - \alpha)$ of the intermediate results in each iteration are distributed over all nodes in the graph. In context of the SkillRank, it is desirable to have $\alpha = 1$ because that would take only the structure of the skill graph into account for the calculation of the credibility ranking. However, the dampening factor also serves the purpose of ensuring that the Markov chain is primitive, which is a necessary precondition for a quick convergence to a well-defined fixed-point solution. The web graph can be considered primitive for all practical purposes [14]. In case of skill graphs, however, there are quite often loops that inhibit primitivity. Also, a choice of $\alpha \approx 1$ usually increases the number of iterations as well as the sensitivity of the solution to small changes in the graph [14]. So the problem is trading off respecting the exact graph structure against fast and guaranteed convergence to the solution as well as a "correct" solution against a "stable" solution. For the skill graphs we have tested so far, a compromise of $0.95 \leq \alpha < 1$ seems to deliver sufficiently good rankings in all cases. This has to be verified by formalized experiments, though.

The third adjustment addresses isolated nodes in the skill graph. These occur when a user's claim to have a skill does not have any confirmations, yet. Obviously, such a node is not connected to the remainder of the graph, a fact that cannot happen for the web graph as it is built strictly by following hyperlinks. For a skill graph, all users having a skill are taken into consideration, including unconfirmed claims. The resulting disconnected nodes, however, cannot be included in the SkillRank computation and are assigned a SkillRank value of 0. This is quite intuitive as the claim obviously cannot be confirmed within the social network and, thus, is not credible. In the example from Fig. 2, user 8 is such an isolated node.

In combination, the confirmations mechanism and the SkillRank algorithm provide a novel approach to online skill profiles that addresses the requirements set forth in Sect. 3, which is made evident in Table 1. Based on these results, a prototype implementation of the approach is sketched in the next Section.

Table 1 Fulfillment of the requirements by the new skills profiles approach

Req.	Fulfillment by the proposed approach
1	Deliberate manipulations by a single user are inherently impossible because of the confirmations approach. Multiple users cartelizing to upvalue their profiles are addressed by the SkillRank which ought to report low credibility for such claims
2	Revenge evaluations are impeded by the confirmations approach because a malicious user cannot take any action except intentionally *not* confirming an experience he or she knows to be true. Offending remarks are filtered out by the profile owner because he or she has to acknowledge every received confirmation before it is published
3	Due to scarce information provided by the OSo API, this requirement has not yet been addressed by the prototype described in Sect. 5. In general, however, the approach can take into account that two users have similar background (e.g., worked in the industry or for the same company) or that they have been reliably confirmed for the skill in question
4	Due to insufficient information in the social graph of our partner from practice (which represents the typical information a social network provider has), the only clue we can use to decide on the evaluator's closeness to the evaluated user is whether they are direct contacts in the social network. If they are, it is supposed that they can evaluate one another fairly accurately, otherwise, they may not confirm each other's experiences
5	The approach allows the user to enter any free text he deems appropriate for a skill descriptor. This allows for highest flexibility at the price of having synonyms or spelling variants in the database. As req. 6 generally does not allow a moderator to build an ontology or curate these variants, we suggest an auto-suggest mechanism that suggests the most popular spelling variant to the user when such a situation is detected
6	Apart from the typical systems administration, occasional user support and routine checks on the application, there is no specific need for human intervention in our approach. Still, it is recommended to have dedicated staff who can set examples of good profiles, answer questions or spot unforeseen problems in the system

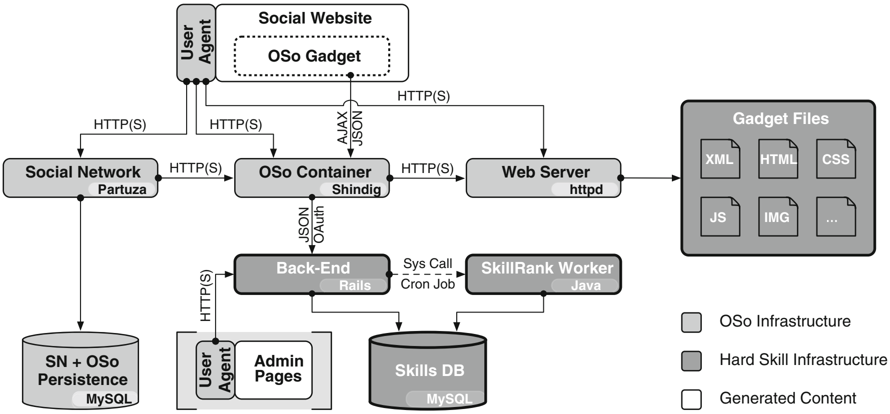


Fig. 3 Architecture of the reference implementation within a large European social network

5 Implementation of the Conceptual Model

In order to test the theoretical concepts discussed above, a skills profile prototype has been implemented based on OpenSocial, Ruby on Rails and Java. OpenSocial (OSo) is an API that allows social network websites to incorporate portable programs, so-called *gadgets*. We chose OSo for the skills profile gadget mainly because of the potential portability that theoretically allows the application to be "plugged into" any OSo-enabled social network. Ruby on Rails and Java are used in the backend server for the more complex calculations.

An OSo gadget is defined by an XML file that usually contains or references JavaScript programs and HTML/CSS contents. The gadget is usually provided as an $<$ `iframe` $>$ by the OSo container, for which we use the reference implementation *Apache Shindig*. As the social network, we use the bare-bones social network *Partuza* that serves exactly the purpose of experimenting with OSo gadgets in a social network.[1] As the Web server, we chose the Apache HTTP Server. This part of the system architecture is required for all OSo gadgets and does not need to be modified when programming new gadgets. The corresponding elements in Fig. 3 are shown in light gray shading, generated Web pages (the "GUI") are shown in white.

The front-end is supported by a back-end server based on Ruby on Rails that takes care of data storage and the more complex parts of the application logic (cf. Fig. 3 for an architectural overview). Requests to the back-end are proxied by the OSo container and secured using OAuth. The Rails back-end also starts the SkillRank calculation whenever necessary. The SkillRank implementation is a multithreaded Java program built on the Colt high-performance computing library. It uses the iterative algorithm from [11] to calculate the SkillRank values for all users and skills in an efficient

[1] For the final release, Partuza was replaced by the social network software of our research partner.

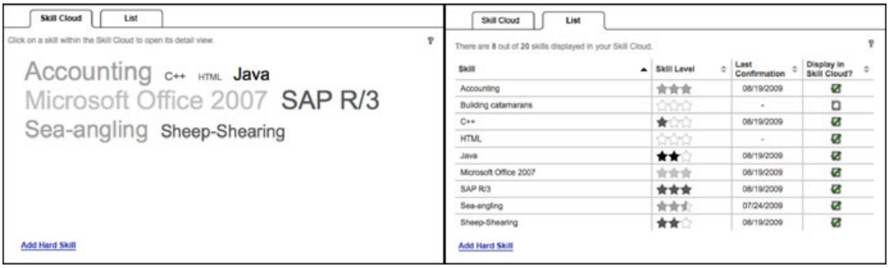


Fig. 4 Screenshots of a sample Skill Cloud (*left*) and the corresponding list view (*right*)

manner. Even large skill graphs can be calculated in a matter of seconds.[2] The parts provided specifically for the skills profile gadget are shown in dark gray shading in Fig. 3.

The gadget allows a user to create an individual skills profile which is displayed as a table or *Skill Cloud,* i.e., similar to a tag cloud (see screenshots[3] in Fig. 4). The size of the skill in the Skill Cloud represents the user's level of expertise for that skill while the intensity of the color indicates how credible the entry is according to the SkillRank calculation. The Skill Cloud is an important tool to get a quick overview of a person's skills profile. The list view presents another perspective on the user's skills including additional information like the date of the last confirmation and the option to publish or hide a skill.

6 Limitations and Future Work

The proposed approach has been adequately grounded on ample theory. First informal tests by a small group of users have been very positive and promising. A larger test phase in a large real-live social network will be conducted during the following months.

While the confirmations approach held up very well in the first tests, the SkillRank needs more investigation. A viable way to render it more intuitive might be to use "credibility brackets" where an unknown person U is enclosed between two known persons A and B, such that $A \prec U \prec B$ (where "$\prec$" means "is less credible than"). This allows the viewer a better judgment of U's absolute credibility because he can compare it with two instinctively familiar credibility scores.

A second limitation of the SkillRank is that credibility does not apply to the skill level, but only to the skill as such. Currently, careful planning a skills profile can

[2] A reasonably sized skill graph of 10,000 users with around 200,000 confirmations is computed in slightly more than one second on an average quad-core PC.

[3] These "screenshots" have been defaced at the request of our research partner from practice, yet the fundamental concepts are unchanged.

lead to a situation where both the average skill level as well as the credibility is lifted to an unduely high figure. This situation is hard to replicate even in small groups, but additional experiments have to show how much influence users can have on their credibility result.

The choice of OpenSocial also imposes technical limitations. While the implementation has proven viable, the application has outgrown the typical size of an OSo gadget. Many parts of the algorithm access the base data in an intensity for which the OSo API is not designed. Moreover, some important features require data not provided by the OSo API which forced us to extend it in some places.

The gadget has been deployed for general use at our research partner from practice so that we are able to collect real-world data which will be invaluable for future evaluation of the approach. Apart from evaluating the approach in its current state, we also would like to extend our research into other directions, including the following:

- What additional information can be incorporated into the credibility calculation to make it more robust?
- How can we take into account the aging of old experiences and the different "half-life periods" for various skills?
- How good is the user acceptance for the prototype application?

7 Conclusion

In this paper, we have presented a novel approach to online skills profiles based on user confirmations and the SkillRank credibility ranking. The approach allows a completely unsupervised presentation of rated skills in a social network, including a statement about the user's alleged proficiency in a skill as well as the credibility for that claim. We have motivated it based on the deficiencies of existing rating schemes and have presented a thorough description of the conceptual model, arguing about its plausibility. In addition, a prototype implementation was presented and current limitations as well as future research were outlined. Due to space limitations we have concentrated on describing the core of the approach and masked out related issues such as the automatic identification of identical skills, additional incentives for the users to use the gadgets etc. In conclusion, we have set forth that our approach is conceptually sound and works from a theoretical point of view. Preliminary user responses confirm this claim. As next steps in our research, we are going to gather more comprehensive empirical data and overcome existing limitations in the approach.

References

1. Bolton GE, Katok E, Ockenfels A (2004) How effective are online reputation mechanisms? An experimental investigation. Manag Sci 50(11):1587–1602
2. Bomze I, Gutjahr W (1995) Estimating qualifications in a self-evaluating group. Qual Quan 29(3):241–250

3. Boyd DM (2008) None of this is real: identity and participation in friendster. In: Karaganis J (ed) Structures of participation in digital culture. Social Science Research Council, New York, pp 132–157
4. Chen KY, Hogg T, Wozny N (2004) Experimental study of market reputation mechanisms. In: Proceedings of the 5th ACM conference on electronic commerce. ACM, New York, NY, USA, pp 234–235. doi:http://doi.acm.org/10.1145/988772.988810
5. Cheung MY, Luo C, Sia CL, Chen H (2007) How do people evaluate electronic word-ofmouth? informational and normative based determinants of perceived credibility of online consumer recommendations in china. Proceedings of the 11th Pacific Asia conference on information systems
6. Conte R, Paolucci M (2002) Reputation in artificial societies: social beliefs for social order. In:Weiß G (ed) Multiagent systems, artificial societies, and simulated organizations, vol 6. Springer, New York
7. Dellarocas C (2001) Analyzing the economic efficiency of ebay-like online reputation reporting mechanisms. In: Proceedings of the 3rd ACM conference on electronic commerce. ACM, New York, NY, USA, pp 171–179. doi:http://doi.acm.org/10.1145/501158.501177
8. Dellarocas C (2003) The digitization of word of mouth: promise and challenges of online feedback mechanisms. Manag Sci 49(10):1407–1424. doi:http://dx.doi.org/10.1287/mnsc.49.10.1407.17308
9. Efimova L (2004) Discovering the iceberg of knowledge work: a weblog case. In: Proceedings of the 5th European conference on organisational knowledge, learning and capabilities. URL https://doc.telin.nl/dscgi/ds.py/Get/File-34786/OKLC_Efimova.pdf
10. Griffith RL, Chmielowski T, Yoshita Y (2007) Do applicants fake? An examination of the frequency of applicant faking behavior. Pers Rev 36(3):341–355. doi:http://dx.doi.org/10.1108/00483480710731310
11. Haveliwala TH (1999) Efficient computation of pagerank. Technical report 1999-31, Stanford InfoLab. URL http://ilpubs.stanford.edu:8 0 90/3 8 6/
12. Jsang A, Ismail R, Boyd C (2007) A survey of trust and reputation systems for online service provision. Decis Supp Syst 43(2):618–644. doi:http://dx.doi.org/10.1016/ j.dss.2005.05.019
13. Kelleher T, Miller BM (2006) Organizational blogs and the human voice: relational strategies and relational outcomes. J Comput Med Commun 11(2):395–414
14. Langville AN, Meyer CD (2006) Google's pagerank and beyond: the science of search engine rankings. Princeton University Press, Princeton
15. Page L, Brin S, Motwani R, Winograd T (1999) The pagerank citation ranking: bringing order to the web. Technical Report 1999-66, Stanford InfoLab. URL http://ilpubs.stanford.edu:8090/422/. Previous number = SIDL-WP-1999-0120
16. Peters R, Reitzenstein I (2008) Reputationssysteme im eCommerce-Funktionsweise, Anwendung und Nutzenpotenziale. HMD—Praxis der Wirtschaftsinformatik 45(261):43–50
17. Resnick P, Zeckhauser R (2001) Trust among strangers in internet transactions: empirical analysis of ebay's reputation system. Working paper for the NBER workshop on empirical studies of electronic commerce
18. Resnick P, Kuwabara K, Zeckhauser R, Friedman E (2008) Reputation systems. Commun ACM 43(12):45–48. doi:http://doi.acm.org/10.1145/355112.355122
19. Schmidt FL, Hunter JE (1998) The validity and utility of selection methods in personnel psychology: practical and theoretical implications of 85 years of research findings. Psychol Bull 124(2):262–274
20. Vossen G, Hagemann S (2007) Unleashing Web 2.0—from Concepts to Creativity. Morgan Kaufmann, San Francisco
21. Winkelmann A, Herwig S, Poppelbuß J, Tiebe D, Becker J (2009) Discussion of functional design options for online rating systems: a state-of-the-art analysis. In: Proceedings of the European conference on information systems. Verona, Italy

Better Support for User Participation Using Business Rules Approach?

Nicklas Holmberg and Odd Steen

Abstract User participation in requirement analysis (RA) is necessary for IS quality and user acceptance. A prerequisite for meaningful user participation is that the coming users also understand the requirements. This understanding is made difficult by abstract and "technical" modelling languages and notations which require learning and experience. The Business Rules Approach (BRA) builds on a notion of Business Rules (BR) formulated in natural language sentences aimed at the business audience; hence BRA should make user participation easy. This is tested in a workshop with a vaccination expert (VE) in a project on designing a BR oriented, digital service for health care workers (HCWs). The results indicate that natural language BRs in RA really are easily understood and intuitive for the VE and that quality checking BRs requires no special learning.

1 Introduction

Conceptual modelling means to define and formally document user requirements on an information system (IS). The conceptual models are used to develop information systems that meet the user requirements and are central to IS analysis and design [15]. The conceptual models are a means for communication of requirements among the different stakeholders in an IS project. Thus, conceptual modelling is central to requirement analysis (RA).

Conceptual modelling in RA is normally done using "technical" and diagrammatical languages such as ER, EER, UML and EPC to capture and define e.g. business concepts and processes.

Many more or less different such modelling languages exist. Moody list 40 conceptual approaches or modelling languages in [15] but conclude that only 20% of these are empirically tested. The lack of empirical research on conceptual modelling,

N. Holmberg (✉)
Department of Informatics, Lund School of Economics and Management,
Lund University, Lund, Sweden
e-mail: nicklas.holmberg@ics.lu.se

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_24,

in this case process modelling, is also discussed in [12], while in [13] it is stated that the research interest in process modelling is formal rather than practical. Concerning ER and Class modelling, Hitchman [8] conclude that there is a lack of research on the effects of different notations in practical modelling.

Thus, several authors point to the need for more practical and empirical research into the use of modelling notations and languages. This would be important since conceptual modelling, as stated above, is important for communication of requirements among stakeholders and therefore also for user participation in RA. However, it is thus also important that the conceptual models are understandable to the involved stakeholders and IS designers [6], or, as Jagielska et al. [9] write,

> One of the a priori conditions for a stakeholder's participation in determining the requirements for a software development project is their ability to fully understand what is being talked about. (p. 914, italics in original)

The problem of direct user participation in RA work with the use of modelling languages and notations has been highlighted especially in Interaction Design and Human-Computer Interaction. In these fields different approaches for easier developer-user communication, such as storyboarding, mock-ups, scenarios, prototyping, etc. have been devised (see e.g. [5, 18]).

In the field of conceptual modelling, the problem of the users understanding the models and diagrams have also been noted. Masri et al. [11, p. 36] conclude that. "... ER diagrams often pose a significant challenge for users even when presented in their own native language." and that reading ER risk placing a cognitive overload on the user, especially for those users who are unfamiliar with the precise semantics of ER. Understanding the normative language of the modelling approach is required to understand the modelling diagrams [8]. Moody [14] studied graphical ER models and found them to be hard for users to understand because of e.g. the technical "look" and abstract nature of ER diagrams.

One of the pioneering Scandinavian participation project, UTOPIA [21], found problems with user participation and involvement because of the abstract nature of the systems descriptions used for communication with the to-be users [2].

Thus, even the limited empirical research into practical understandability of conceptual modelling languages and the effect on user participation in RA suggest that technical and constructed languages, such as ER and process modelling notations, creates difficulties for non-experts.

In recent years, a new modelling paradigm has gained ground, namely the Business Rules Approach (BRA). BRA focuses business rules (BR) in RA and ISD as the third essential IS pillar next to business concepts and business processes.

One of the essences in the BRA is that rules should be written declarative in as natural language as possible and that BR should be managed by business and not IT experts (see e.g. [7, 16, 26]). The BRA thus implicitly promises and presupposes that the BR modelling notation and language can be so natural that to-be users not only understands the BR models during RA, but also are able to maintain and modify the BRs of the implemented IS. If this would be the case, the BRA has finally provided the ISD practice with a truly user-understandable technique for RA (albeit for the

BR part of an IS). BRA should thus place a lesser cognitive burden upon the users and participants.

This paper aims to give provisional answers to whether BR modelling using structured but natural written language following Tony Morgan's [16] templates for BR formulation is easy and intuitive enough to work in RA with user participation without special training and learning of techniques and notations. To the best of our knowledge, this has not before been reported based on empirical research.

The provisional answers are based on early findings from modelling a BR oriented digital service called VacSam. The VacSam project[1] aims to provide health care workers (HCWs) with an automated decision support for diagnosis of immigrant children's vaccination compared to the Swedish vaccination schedules for children, and subsequently for individual vaccine prescription recommendation based on the diagnosis.

The remainder of this paper is structured as follows. The next chapter discuss the problem of immigration and child immunization, and describe the VacSam project and digital service. The third chapter explains Business Rules and design of these and the Business Rules Approach. The fourth chapter presents the empirical investigation. We end the paper with our conclusions.

2 People Mobility and Immunization

In 2008 there were 120,000 children[2] born in another country but living in Sweden [20]. Also in 2008, 45,294 persons[3] emigrated from Sweden to return to old home countries or to move to other countries [21]. This people mobility creates problems for child immunization.

Each country in the world has its national immunization schedule, in which vaccines, vaccination intervals, and doses are set. The various national schedules are published by WHO on the web for public access.[4]

An excerpt of this spreadsheet is shown in Fig. 1, depicting the national immunization schedule of Sweden. When we compare this schedule to Finland's[5] in the same WHO file (Finland's schedule is not shown here) many differences emerge, such as:

1. Sweden has a hexavalent combination vaccine in DTaPHibHepBIPV, but in Finland HepB is a separate vaccine.
2. Sweden has Pneumo_conj vaccine. Finland does not.
3. Finland has HepA, MenACWY, Rotavirus, TBE, and Td vaccines, which Sweden does not have.

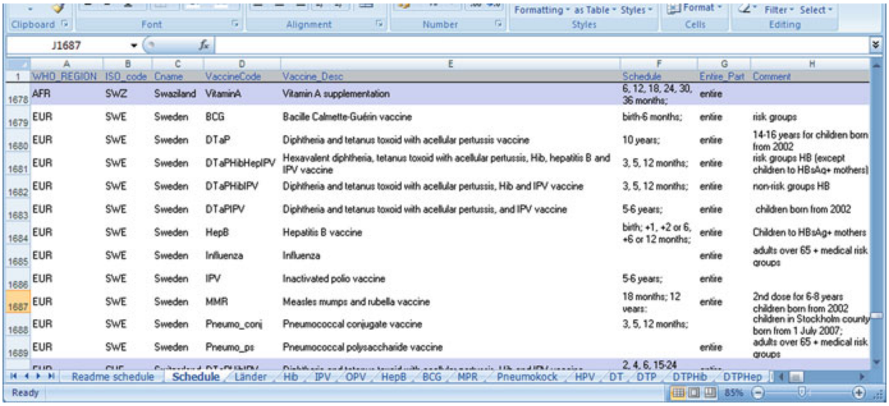
[1] http://www.ics.lu.se/en/research/projects/vacsam.

[2] A child here is a person in the age of between 1 and 17 years.

[3] Separate statistics for children is not available.

[4] http://www.who.int/immunization_monitoring/en/globalsummary/ScheduleResult.cfm.

[5] A country, which share borders and have a long and common history with Sweden, and which in many ways, could be considered as quite similar to Sweden.



	WHO_REGION	ISO_code	Cname	VaccineCode	Vaccine_Desc	Schedule	Entire_Part	Comment
1678	AFR	SWZ	Swaziland	VitaminA	Vitamin A supplementation	6, 12, 18, 24, 30, 36 months;	entire	
1679	EUR	SWE	Sweden	BCG	Bacille Calmette-Guérin vaccine	birth-6 months;	entire	risk groups
1680	EUR	SWE	Sweden	DTaP	Diphtheria and tetanus toxoid with acellular pertussis vaccine	10 years;	entire	14-16 years for children born from 2002
1681	EUR	SWE	Sweden	DTaPHibHepIPV	Hexavalent diphtheria, tetanus toxoid with acellular pertussis, Hib, hepatitis B and IPV vaccine	3, 5, 12 months;	entire	risk groups HB (except children to HBsAg+ mothers)
1682	EUR	SWE	Sweden	DTaPHibIPV	Diphtheria and tetanus toxoid with acellular pertussis, Hib and IPV vaccine	3, 5, 12 months;	entire	non-risk groups HB
1683	EUR	SWE	Sweden	DTaPIPV	Diphtheria and tetanus toxoid with acellular pertussis, and IPV vaccine	5-6 years;	entire	children born from 2002
1684	EUR	SWE	Sweden	HepB	Hepatitis B vaccine	birth; +1, +2 or 6, +6 or 12 months;	entire	Children to HBsAg+ mothers
1685	EUR	SWE	Sweden	Influenza	Influenza		entire	adults over 65 + medical risk groups
1686	EUR	SWE	Sweden	IPV	Inactivated polio vaccine	5-6 years;	entire	
1687	EUR	SWE	Sweden	MMR	Measles mumps and rubella vaccine	18 months; 12 years;	entire	2nd dose for 6-8 years children born from 2002
1688	EUR	SWE	Sweden	Pneumo_conj	Pneumococcal conjugate vaccine	3, 5, 12 months;		children in Stockholm county born from 1 July 2007;
1689	EUR	SWE	Sweden	Pneumo_ps	Pneumococcal polysaccharide vaccine		entire	adults over 65 + medical risk groups

Fig. 1 Excerpt showing Sweden in WHO's spreadsheet of all the vaccination schedules in the world

Thus, when an immigrant child of a certain age from e.g. Finland should be immunized according to the Swedish schedule, the Swedish healthcare system face two determining questions:

1. What vaccines does the child lack according to the Swedish schedule?
2. What vaccines should the child have been given according to the schedule of the country the child came from?

To answer these questions, HCWs must thoroughly know the Swedish schedule and the schedule of the child's previous country. They must also know the immunization history of the child. These requirements are however hard to meet.

The effect of this is over use of vaccines. Immunization experts in the VacSam project informally estimate that 70–80% of all vaccinations in Sweden are unnecessary.[6] This results in unnecessary suffering for children with increased risks of contraindications and allergic reactions, and a waste of taxpayers' money and medical resources. What is needed is a help for the HCWs to avoid over use of vaccines and increase the quality and control of immunizations in Sweden. The VacSam service will be this help.

The immunization practice is highly regulated through the various schedules, which suggest using Business Rules Approach. Another consideration is that it is important that the rules governing the immunization process belong to the practice and not the IT department, so that changes in regulation (especially in the schedules) can be implemented in the service by designated HCWs who are business experts rather than experts of IT. This is one of the cornerstones of the Business Rules Approach.

Even if the rules in the form of immunization schedules and other vaccination regulations are not very volatile or in constant flux, the argument for a business rules oriented digital service is still strong. The immunization practice is a business where exact and uniform operations and explicit conformance to regulations are vital. The

[6] According HCWs, the pinpricks should be kept to a minimum.

problem is that these requirements are presently not fulfilled—a problem that can be remedied through a business rules oriented IS in the form of a digital service.

3 Business Rules Oriented IS Design

BRs should be based upon facts; in turn, facts should be based upon concepts that are represented by terms. By motivating the BRs through important aspects of the business, BRs can affect the behaviour of the organization in a desired direction [20]. BRs should be available for business experts and have only one source through the BR repository. The rules should also be specified by defendant personnel and they have to be manageable [20]. Morgan [16] presents a definition of a Business Rule:

> [. . .] the conditions under which a process is carried out or the new conditions that will exist after a process has been completed. [16, p. 59]

A BR can be interpreted as a claim which defines or delineates an aspect of a business [24].

3.1 The Business Rules Approach

BRA is a systems development approach, which accommodates the key entities of the BR movement. BRA is neither the latest nor the best system-development methodology. It does not replace the requirement analysis but is not just another set of tools [25]. BRA has the quality of isolating the effects, which are brought by changing business logic. Also, when a change occurs, the approach leads to that the business is only affected to a controllable extent [23] by the ease of BR management in natural language which is provided by the central component of BRA: BRMS.

BRA has developed through Information Systems Science from both the professional and the academic domain. BRA advocates the usage of the core components of what can be found as the BR paradigm, including that BRs are expressed in "natural language" that can be executed by an Information System (IS).

The basic principles of BRA accommodates that BRs always should be explicitly expressed in a natural language. Also, BRs should exist independently from workflows and procedures [20]. BRA is an approach that holds the power of resetting the influence of the business logic/BR from the "IT Department" to the defendant business personnel. The "IT Department" though should still be responsible for the technicalities corresponding to the implementation of the Business Rules Management Systems (BRMS) or other technicalities associated with BRA [7, 16].

BRA includes BR design that excludes expressions corresponding to "how" the rule is executed, "where" the rule is executed, "who" is responsible for the rule execution and "when" the rule is executed. An example could be stated as:

> A society has to include both members and non-members.

This rule does not state "how", "where", "who" or "when" and is therefore accepted. It is also presented in a declarative manner [20]

The BR management should be executed as an independent discipline of systems and business development within BRA [26]. The discipline could thus advocate clarification of the BR. The application of BRA could in turn lead to that, the BR will be implemented as well-formed when following recommendations and guidelines provided by i.e. BR-Manifesto.

Dedicated techniques such as BRMS, Business Rules Engine (BRE) and BR repository are other key components within BRA. The application of them leads to a separation of BR from application specific code [1, 7].

BR repository enables the use of BR management, with accompanying documentation, as a strategic resource [4, 16]. This is because the BR repository enables the quality, versioning, traceability, and responsibility control of the Business Rules. A BRE provides an automated rule application of rule set(s) and rule flow(s) on which inference such as backward and forward chaining is enabled.

BRA strengthens thus the relations between the business goals and visions, the implemented BR and the anchorage of the BRs in the Enterprise Model [1, 17, 19].

3.2 Business Rules Design and Development

According to Morgan [16] BRs holds a structure but allocates different parts of the meaning with the business and the desired business properties. Below is an example of how BRs could be presented on different levels of representation [16].

At the least formal level (the business level) a BR would be expressed in natural language in a limited frame of patterns and look like:

> A credit account customer must be at least 18 years old. [16, p. 63]

At the technical level a BR is a combination of structured data references which limits the natural language:

> CreditAccount Self.customer.age >= 18. [16, p. 63]

At the most formal level a BR is expressed in a defined syntax that holds certain specific mathematical properties:

> {X, Y, (customer X) (creditAccount Y) (holder X Y) ⇒ (ge (age X) 18)}. [16, p. 63]

To achieve the highest grade of structure, the most formal level of expression would be to consider. However this would probably lead to a hard time for business experts figuring out what the BR means [16]. According to the BR manifesto:

> Rules should be expressed declaratively in natural language sentences for the business audience. [3, Article 4.1].

The condensed explanation of how to develop BRs is by analyzing the business. Most BRs could probably be found by performing a business analysis. The focus would be on the business plan but could also be on documentation working as strategic benefits in this phase. The BRs adds the ground for the business vocabulary, which is derived from BRs. The vocabulary illustrates the relationships between terms in the ontology which accommodates the Swedish vaccination business, e.g.:

Child *is a* Person, Foreign Child *is a* Person.

This vocabulary is derived from the business plan (i.e. The National Board of Health and Welfare regulations) and BRs using these terms in our case looks like:

RS-1. A person is defined as a child if all of the following conditions are true:

the person is a new born
the person is at most 18 years old

RS-2. A child is defined as a foreign child if the child comes from another country than Sweden.

BRs like these were quality checked with a vaccination expert (VE) during design of the BR model, which is discussed below.

4 Experiences from User Participation in Business Rules Oriented IS Design

We have followed the advices given in [7, 10, 16] about where to look for business rules. We have performed a static analysis [16], i.e. gone through and analysed documentation in the form of regulations published by The National Board of Health and Welfare [22], policies and guidelines from Region Västra Götaland and Stockholms läns landsting,[7] the WHO schedules spreadsheet [27], and reports on immunization in Sweden from SMI.

Based on these texts and documents we have designed BRs in natural Swedish, according to the guide lines on rules formulation found in [16, 20]. In this text, the BRs are translated to English.

4.1 Rule Analysis Workshop

One phase in the design of the BR centric VacSam service is the quality check of the business rules. This task is carried out through workshops with "business owners/personnel". The workshop aimed to analyze BRs which are based on the Swedish vaccination schedules. The workshop consisted of six stages and was based upon "analysis workshop" in [16].[8] The third stage was carried out with one of Sweden's leading vaccination experts and was video recorded as a mean to collect data.

[7] The two largest country councils in Sweden.

[8] (1) Define goal and approach; (2) Prepare for workshop; (3) Conduct the workshop session; (4) Pursue immediate follow-up activities; (5) Follow up with consolidation and research; (6) Review.

The purpose of the workshop was to quality check the, by that time, 70 BRs acting governing over the vaccination process. The workshop focused on changing, understanding, removing, adding, and prioritizing the BRs. The workshop was semi-structured to give room for questions and dialogue outside the frame of the workshop.

The video recorded session was transcribed and coded using the following codes:

- Change = C: Sentences that focused on changing a BR, taking business perspective and specific vaccination offers, into account.
- New = N: Sentences that focused on new BR.
- Removal = R: Sentences that focused on removing BR, taking the control of constraints and regulations and following rules into account.
- Priority = P: Sentences that focused on the priority of BR, as expressed by the expert.
- Understanding = U: Sentences that focused on the understanding of the vocabulary and understanding of the declarative, atomic business rules.

4.2 Findings from the Workshop

The VE who was taking part in conducting the analysis workshop in the design phase and the quality control of the BR that governs the vaccination activity we could find that VE could figure out what the BR meant.

In the example above, where the business ontology is derived from the BR, RS-1 corresponds to the definition of a child given by the The National Board of Health and Welfare.

VE commented:

> yes, well, you are a child from the moment when you are born and the breakpoint is often at eighteen years of age when the authority rule becomes effective (RS-1. U, C).

Rule statement 8:

> RS-8: A child should be given separate vaccines against diphtheria and tetanus, if the child is at least 12 years old.

VEs direct response to this BR was:

> There you will have to add if it is a basic vaccination (RS-8. C, P).

By the input from VE we could change and thereby also prioritize RS-8 to be in line with the Swedish vaccination schedule:

> RS-8: A child should be given separate vaccinations against diphtheria and tetanus, if all of the following conditions are true:
>
> the child is at least 12 years old
> the vaccinations are basic

- RS-17 could further exemplify VEs understanding of RS 1-70.

> RS-17: A child should be given 1 dose of vaccine against Hemophilus Influenzae type B, if the age of the child is over one year.

- The direct response from VE was:

> I just want to make sure that [checks the regulation texts], this is a matter of age, a movement of age, no it is not in here, no okay, that is correct! (RS-17. U).

- Note that the question for VE was never about the understanding of the language used to present the BR, but rather if the BR were in line with the present Swedish vaccination regulations. Further notice of ours, to denounce the above stated, is BR

> RS-32: A risk child should be given vaccine against PPD, if the Child arrived to Sweden three months ago.

- VE responded:

> This is totally incomprehensible! (RS-32. U).

- However, it was once again not the BR that was hard to understand, neither a question of the understanding of the vocabulary derived from the ontology, but rather if RS-32 was in line with the Swedish vaccination regulations. VE continued:

> I think you have mixed two things here; PPD is a test you do for getting to know if the child is vaccinated or is sick with tuberculosis, BCG is the vaccine against tuberculosis (RS-28. C, U).

- This statement, coded Change and Understanding, resulted in a change of RS-32:

> RS-32: A risk child should be given BCG vaccine against tuberculosis if the child arrived to Sweden three months ago.

- The priority of the BR was another interesting observation. BR 1-70 which were presented to VE all relates to each other. When all the rules are true, a child is fully vaccinated according to the Swedish vaccination schedule. At the end of the workshop VE found that the order in which the rules were presented was not logic and stated:

> It would have been easier if you had written those [number of BR] first.

- Further exemplifying the understanding of RS 1-70 was small details within specific rules which led to change, new rule, deletion and addition. Such details could be:

> RS-43: A child should be given three doses of Diphtheria, Tetanus, Pertussis, Polio and Hib combination vaccine, if the child is at least five months old.

- VE directly commented:

> Dose three. This meant not three doses but the third dose (RS-43.U, C).

Hence, we changed RS-43 into:

> RS-43: A child should be given dose 3 of Diphtheria, Tetanus, Pertussis, Polio and Hib combination vaccine, if the child is at least five months old.

However, the interaction with VE was never a question of ontological misconceptions neither a question of the understanding of the vocabulary but rather if the BR were correct or not according to the Swedish regulations.

However the case might have been different if the BR were presented at a more formal level corresponding to an imperative paradigm i.e. C#.

It is also worth mentioning that VE could recognize a BR which we had derived from the regulations of the National Board of Health and Welfare regulations (SOSFS) and FASS[9] which was something that we had not used at all. In fact that was part of what VE was scrutinizing and said:

> I just want to make sure that, this is a matter of age, a change of age, **no it is not in here** [FASS + SOSFS, i.e. regulatory texts], no okay, that is correct! (RS-17. U).

5 Conclusions

This paper presents early experiences from designing and developing a BR oriented digital service for immunization recommendations for immigrant children in Sweden. The experiences are based on a video recorded BR workshop with one of Sweden's leading expert on immunization and vaccines (the VE). As such the VE is a business expert, but definitely not an ISD or conceptual modelling expert and no previous experience of BR modelling.

What the analysis of the BR workshop show is that the declarative approach providing a representation of the BR in natural language seems to be naturally and immediately understood by the VE. The VE could directly comment what was right, wrong, or in need of change, etc. Thus, the VE could fully focus on the *meaning* of the formulated BR and how they comply with regulatory texts and practice. At no time the VE had to try to understand an abstract or technical language, or the normative language of the modelling approach [8]. Thus, the VE needed no special learning of a modelling language.

We have used the least formal level of BR representation [16] which obviously put a lower cognitive burden on the VE, which strengthens the BRA proponents' position that business experts rather than IT-experts could and should manage and control the implemented BR.

With this study we have provided early and tentative support for the assumption about the ease for business experts to understand BR formulated in natural language. The study is limited to only one business expert and for greater generalisability more empirical evidence is needed, which will be made possible through further studies within the VacSam project.

Acknowledgements The VacSam project is sponsored by VINNOVA.

[9] The Swedish pharmaceutical industry's published list of drugs.

References

1. Bajec M, Krisper M (2005) A methodology and tool support for managing business rules in organisations. Inf Syst 30(6):423–443
2. Bjerknes G, Bratteteig T (1995) User participation and democracy: a discussion of Scandinavian research on system development. Scand J Inf Syst 7(1):73–98
3. BRG (2003) The business rules manifesto. Business Rules Group. http://www.businessrulesgroup.org/brmanifesto. htm. Accessed 23 Apr 2010
4. Ceponis L, Vasilecas O (2006) Logical view of architecture of business rules repository. Informacijos Mokslai 36:139-147
5. Dix A et al (2004) Human-computer interaction. Pearson, Upper Saddle River
6. Gemino A, Wand Y (2005) Complexity and clarity in conceptual modelling: comparison of mandatory and optional properties. Data Knowl Eng 55(3):301–326
7. Graham I (2006) Business rules management and service oriented architecture: a pattern language. Wiley, Chichester
8. Hitchman S (2002) The details of conceptual modelling notations are important—a comparison of relationship normative language. Commun Assoc Inf Syst 9(10):167–179
9. Jagielska D et al (2006) How natural is natural language? How well do computer science students write use cases? In: Companion to the 21st ACM SIGPLAN symposium on object-oriented programming systems, languages, and applications, Portland, OR, USA
10. Kardasis P, Loucopoulos P (2005) A roadmap for the elicitation of business rules in information systems projects. Bus Process Manage J 11(4):316–348
11. Masri K et al (2008) Using iconic graphics in entity-relationship diagrams: the impact on understanding. J Database Manage 19(3):22–41
12. Mendling J et al (2007) What makes process models understandable? In: Proceedings of the 5th international conference on business process management, Brisbane, Australia. Springer, Berlin
13. Mendling J et al. (2010) Seven process modelling guidelines (7PMG). Inf Softw Technol 52(2):127–136
14. Moody D (1996) Graphical entity relationship models: towards a more user understandable representation of data. In: Thalheim B (ed) Conceptual modelling ER'96. Springer, Berlin, pp 227–244
15. Moody DL (2005) Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. Data Knowl Eng 55(3):243–276
16. Morgan T (2002) Business rules and information systems: aligning IT with business goals. Addison-Wesley, Boston
17. Object Management Group (2008) Business motivation model version 1.0. Object Management Group. http://www.omg.org/docs/formal/08-08-02.pdf. Accessed 20 Aug 2008
18. Preece J et al (1994) Human-computer interaction. Addison-Wesley, Wokingham
19. Rosca D et al (2002) Enterprise modelling and decision-support for automating the business rules lifecycle. Autom Softw Eng 9(4):361–404
20. Ross RG (2003) Principles of the business rule approach. Addison-Wesley, Boston
21. Sandberg A (1983) Trade union-orientated research for democratization of planning in work life—problems and potentials. J Occup Behav 4(1):59–71
22. Socialstyrelsen (2008) Socialstyrelsens föreskrifter om ändring i föreskrifterna (SOSFS 2006:22) om vaccination av barn. Socialstyrelsen, Ändringsföreskrift, SOSFS 2008:7 (M). http://www.socialstyrelsen.se/sosfs/2006-22/Documents/2008_7.pdf
23. Thi Anh-Duong H, Thanh Binh N (2009) State of the art and emerging rule-driven perspectives towards service-based business process interoperability. In: 2009 IEEE-RIVF international conference on computing and communication technologies, Da Nang, Vietnam, pp 1–4
24. van Eijndhoven T et al (2008) Achieving business process flexibility with business rules. In: 2008 12th international IEEE enterprise distributed object computing conference, Munich, Germany, pp 95–104

25. von Halle B (1997) The art of letting go. Database programming and design archives. http://www.dbpd.com/vault/9702arch.htm. Accessed 15 Apr 2009
26. von Halle B (2002) Business rules applied: building better systems using the business rules approach. Wiley Computer Publishing, New York
27. World Health Organization (2010) Immunization surveillance, assessment and monitoring. http://www.who.int/immunization_monitoring/diseases/en/. Accessed 17 Mar 2010

From Organization Business Model to Information System: One Approach and Lessons Learned

Janis Iljins and Maris Treimanis

Abstract The system ISTechnology and lessons learned of its use in the business modeling and development of information systems are analyzed in the paper. The system consists of the meta-model and applications. The meta-model enables to define a platform independent business model of the organization. The applications provide the definition and interpretation of the business model. Interpretation of the business model provides functionality of the information system in the selected platform. The lessons learned confirm that the development and maintenance cost of information systems can be significantly reduced by use of the ISTechnology.

1 Introduction

Many approaches are used to carry out business modeling and information systems (IS) development. The ISTechnology is one of them. The first version of the ISTechnology was launched more than 14 years ago. Till nowadays the ISTechnology was successfully used in the development of various information systems [1, 2, 3] and is continuously improved. The goal of the present paper is to explain ideas of the ISTechnology and present the results of empirical research to show the practical benefits of the proposed approach.

The first chapter of the paper describes the most widely used ideas, approaches and tools for business modeling and IS development. Some of the abovementioned ideas can in integrated way be found in the ISTechnology. The second chapter gives a concise description of the meta-model and the applications of the ISTechnology. The lessons learned and benefits from the use of the ISTechnology in business modeling and IS development are presented in the third chapter.

J. Iljins (✉)
University of Latvia, Raina bulv. 19, 1586, Riga, Latvia
e-mail: janis.iljins@lu.lv

J. Pokorny et al. (eds.), *Information Systems Development,*
DOI 10.1007/978-1-4419-9790-6_25,

2 Related Work

The aim of business modeling is to define such a model of an organization, which would be understandable for all stakeholders involved—business and IT experts, end-users, management, etc. [4]. The model is used to analyze, simulate and optimize business of the organization, as well as for more or less automated development of IS.

2.1 Business Modeling Languages and Tools

A lot of universal and domain specific business modeling languages and support tools for business modeling exist nowadays [5, 6, 7]. Many of them, being based on the business model, are able to perform some automation of a software development process. Rational Rose is one of the most popular tools which help to generate software from the model defined in UML language [8]. At the same time, for example, the business modeling language GRAPES-BM and its support tools allow both—to define the business model and to carry out the model simulation [9].

The Intalio|Cloud BPM can be mentioned as one of innovative tools which support the development of the business model and software [10]. This tool supports one of the newest standards for the business model description—Business Process Modeling Notation (BPMN).

There are also some business model based specialized tools, for example, Microsoft BizTalk Server. If web-services are used, there is a possibility to automate execution of these services. For such purposes Web Services Business Process Execution Language [11] and the appropriate tools were developed.

2.2 Standard Software

Various universal and widely used systems are also developed. For example, SAP, Microsoft Navision and many others, which are called standard software by the authors [12]. Enterprises use standard software and describe the business processes according to the models offered by these systems. These systems can also be customized. For example, Microsoft Navision has its own programming language for development of specific functionality of the organization. If the organization chooses standard software, often a lasting and expensive installation process begins, during which—either the functionality of the system or/and the organization business model are changed [13].

2.3 Software Reuse

Also the role of software reuse is significant, for example, Software Product Lines (SPL) techniques can be mentioned [14]. SPL is successfully used in situations, where software development costs are high and safety, and performance are critical according to the authors [15]. One of SPL suggestions is building a family of software products in such a way that you can customize variability for specific requirement sets [16].

2.4 Model Driven Architecture

Model Driven Architecture (MDA) is not a business modeling method. It does not define how to build a business model, but it may be used to get the organization's IS appropriate to the business model. Transformations are used to get one model from other. Transformations may occur in several steps in MDA practice—from general Platform Independent Models (PIM) towards more specific and closer to implementation models (PSM's) until we get a code [4, 17]. According to [4], it is a business model, which can serve as the highest and most general description of the system, which may be further transformed into specific models. MetaEdit+ is the most popular tool, which supports MDA [18].

3 ISTechnology

ISTechnology consists of the meta-model and applications for definition of a platform independent organization business model and implementation of this business model in the specific platform. The ISTechnology, being based on the organization platform independent business model, is compatible with the main ideas defined in [13, 15, 17].

3.1 ISTechnology Meta-Model

According to [19], in order to define an organization business model, one should answer six questions—Why?, What?, How?, Who?, When? and Where?. ISTechnology meta-model allows to answer these questions and also supports recommendations of [20].

The latest version of ISTechnology meta-model consists of 59 object classes and 45 relations. The notion of "Module" is used to improve readability of the meta-model. Each module consists of object classes and relations. The ISTechnology meta-model consists of the Workplace, Organizational, Object, Process, Report and Monitoring modules. The classes and relations of the ISTechnology meta-model have predefined semantic. This semantic is used for automatic documentation purposes of the organization platform independent business model and to provide functionality of the IST Shell application (see Sect. 3.2).

Let us concisely consider the most relevant object classes and relations of the ISTechnology meta-model.

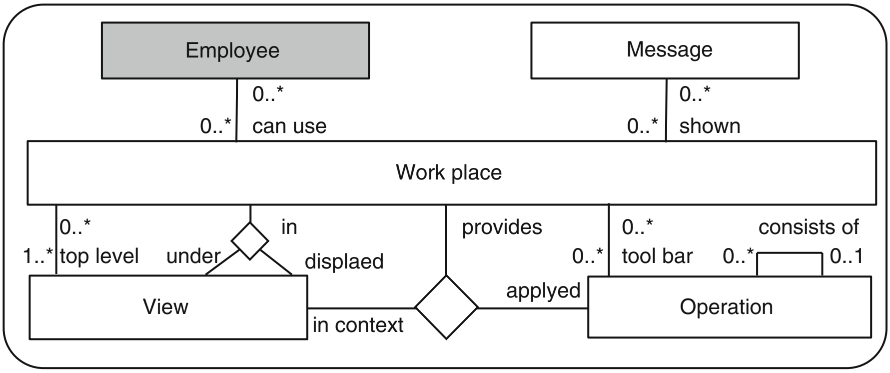


Fig. 1 Workplace module

3.1.1 The Workplace Module

The Workplace Module is used to define the sets of employees' rights and responsibilities in the organization business model (Fig. 1).

The class "*Workplace*" is a placeholder used to define the set of employees' rights and responsibilities.

The class "*Employee*" (see Sect. 3.1.3) and the corresponding relation define employees, which can use the specified workplace. The shaded rectangle means that the object class is defined in the other meta-model module. For example, the object class "Employee" is defined in the "Organizational module".

The class "*View*" and its relations define business objects available in a workplace, which in case of necessity may be structured in views and sub-views.

The class "*Operation*" and the corresponding relations define operations to be executed on business objects in the workplace. As a result of the execution of operations, new objects can be created, the existing ones can be deleted, and attribute values of business objects can be changed. In case of necessity the operations may be hierarchically structured.

The class "Message" and its relation define reminders, which can be viewable in the workplace and indicate that in the workplace according to the business model some operation must be carried out. For example, the reminder "Deal is waiting approval!" is shown in the dealer's workplace, if the transaction of buying a certain amount of some currency is carried out by another dealer.

3.1.2 The Process Module

The Process Module provides definitions of state-transition types of business objects and the corresponding implementation mechanism for object state-transitions (Fig. 2).

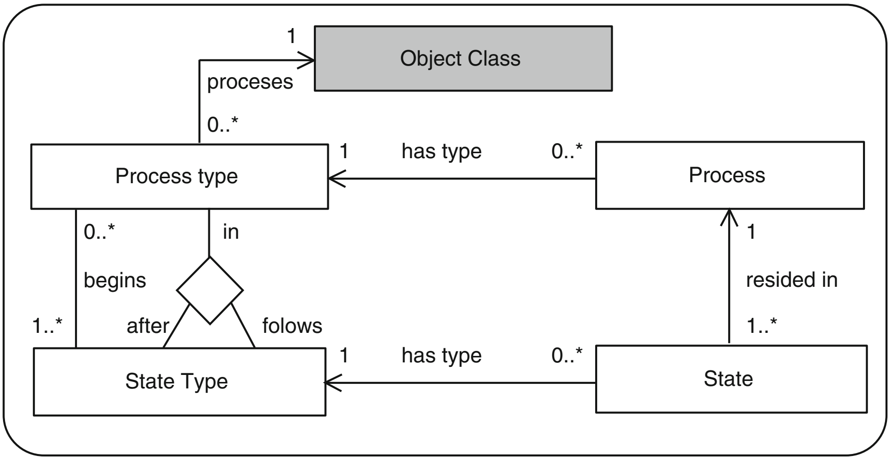


Fig. 2 The process module

The class "*Process Type*" allows to define state transition types of business objects, while the class "*State Type*" defines states of a business object. Relations among the state types identify state transitions of business objects. The binary relation "*Process Type*"–"*State Type*" (cardinality—many-to-many) identify the initial states of the specified process type. The ternary relation "*Process Type*"–"*State Type*"–"*State Type*" identifies the permissible state transitions.

The classes "Process", "State" and the corresponding relations allow executing state transitions of the business object according to the defined process types.

3.1.3 Monitoring Module

The Monitoring Module provides the organization business model monitoring, audit and improvement facilities (Fig. 3).

The class "*Event*" records all operations performed with each Object Class instance during the implementation of the business model. The main attributes of the event:

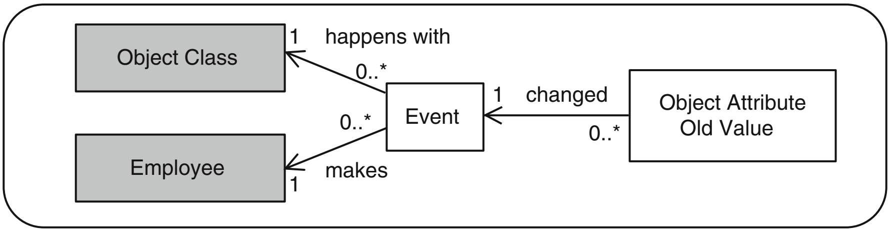


Fig. 3 Monitoring module

- time of the event
- business process type and instance
- object class and instance, by which the operation was performed
- operation—user's entry/exit from the system or viewing, creating, correcting and deleting of the business object
- user, which has performed the operation

The class "*Object Attribute Old Value*" records the values of the previous business object attribute, which was deleted or modified. It should be noted, that not all operations (for example, viewing of attributes) create changes in the values of the attributes.

3.1.4 Other Modules

The Objects Module enables to define the organization business object classes and their relations. This module plays the role of the Object model [7].

The Reports Module enables to define and prepare reports on the defined organization business model and business model dynamics, as well as business domain specific reports.

The Organizational module enables to define organizations, their locations (countries, regions), departments and their organizational structure, as well as employees and their roles. The module enables to define business model components, traditionally used for such purpose. The business model components defined by the organizational module are used to define the authority and responsibility of the employees and departments (Workplace module), as well as for monitoring and analysis of the business model dynamics (Monitoring module).

3.2 ISTechnology Components

ISTechnology consists of IST Shell, domain nonspecific (DNS) and Direpo applications. These applications are included in all IS being developed by ISTechnology. In development of IS, in addition to ISTechnology components, also domain specific (DS) applications have to be developed.

IST Shell application IST Shell application provides the definition and interpretation of the business model according to the IST meta-model. Interpretation of the business model enables to perform operations with business objects. IST Shell provides the "central" user interface of the ISTechnology—window with subwindows and menus. The presentation logic of user interface is defined by the concrete implementation of IST Shell. The user interface enables employees to select objects defined by the business model and perform operations with them. The parameters, appropriate to the context of execution, are transferred to operations in standardized way.

Operations with business objects are performed by help of IST DNS applications, Direpo and DS applications.

IST DNS applicationsn IST DNS applications are reusable and configurable applications, which can be used in all business domains. For example, application for business objects state transitions, which in collaboration with the user, by offering the user permissible state transition option of the given type object, provides transition of the object state received in the parameters. The current version of ISTechnology contains 15 reusable DNS applications.

IST Direpo application IST Direpo application provides preparation of reports according to the definitions established in the business model. The application interprets the report definition, obtains data form the database and formats them in the Rich Text Format document.

IST DS applications DS applications are business domain specific applications, for example, currency buying applications. Within one business domain these applications may be reused directly or customized to specific organization business requirements.

4 Lessons Learned from the Use of ISTechnology

4.1 Objects of Case Study

Let us consider the experience of use of the ISTechnology in the financial business domain (Table 1).

VOIS—ensures currency exchange and deposit deals. Conclusion of buying/selling contracts, control of different limits and registration of currency positions, confirmation of deals, corresponding payments in the SWIFT system and control of the payments are ensured. The processing of the deals is determined by the type and the risk of the deal—either the deal is concluded with the bank or the client of the bank.

Table 1 IS developed by the ISTechnology

Type of information systems	Number of installations in different banks	Year of implementation
VOIS—currency operations IS	2	1995, 1996
PFS—pension fund IS	1	2000
VUS—security accounting IS	3	2002, 2008, 2009
LPS—resources management IS	3	2004, 2007, 2009

PFS—ensures conclusion of contracts with the client, registration of the client pension fund, processing of pay-in/pay-out transactions. Input of the above mentioned data from the outer systems has been automated. Deposits of the participants of the pension fund and the employer of the participant, as well as the profit from investments are registered. Calculation of the taxes from the pensions paid has been automated.

VUS—ensures processing of security deals in the portfolios of the bank or the clients of the bank. Security registration in the client and bank security accounts as well as sending and processing of deals in outer systems (SWIFT, stock exchange, depositary) are ensured. The system fixes the security prices and reevaluates the portfolios.

LPS—ensures investment fund portfolio management. Emitting and clearance of the fund parts, processing of security deals and cash payments, reevaluating of the fund actives and preparation of the balance, determination of the fund part value are ensured. Automated exchange of the deal data with the holding bank security registration system is ensured.

Business processes supported by the above mentioned IS traditionally are carried out by three departments:

- Front-office negotiating and enter deals (for example, buying or selling of currency, security, etc.).
- Back-office ensuring technical transactions of deals (for example, payments, using the SWIFT, etc.)
- Middle-office provides monitoring, control and analysis of the processes within the context of different bank investment portfolios, etc.

IS functionality appropriate to the bank business model is defined by the IST meta-model based platform independent model and IST applications. It should be mentioned that the business model of each bank is specific and, accordingly, IS work places functionality and number of users in each bank are different. IST DS applications were reused directly or modified according to specific requirements of banks. It should be noted, that these applications are characterized by complicated calculation algorithms (calculations of portfolios currency positions, etc.) and non-functional requirements (extensive keyboard use in entering deals, etc.) For development of such applications it is necessary to use universal programming languages.

4.2 Development Effort Analysis

Let us consider the percentage of components of the ISTechnology in the above mentioned IS (Fig. 4).

PFS and LPS IS are functionally simpler, the number of applications is smaller and more than a half (55%) of functionality was provided by use of ISTechnology components. However, also in technically complicated VOIS and VUS IS almost a

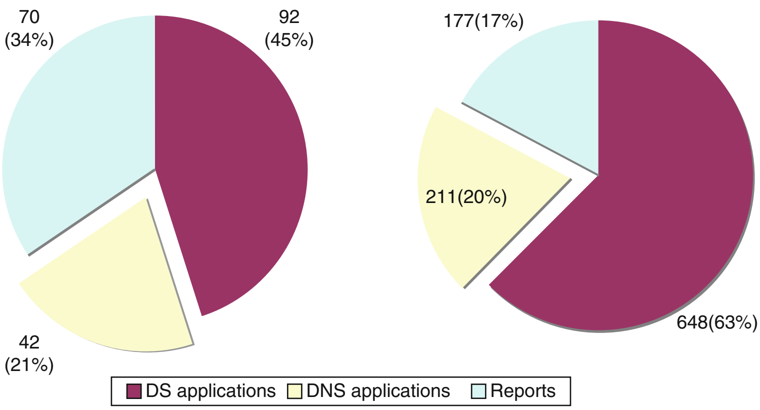


Fig. 4 The percentage of components in PFS and LPSIS on the left and in VOIS, VUS IS on the right

Table 2 Estimated effort

Development activity	Estimated effort[a]
Changes in business model by IST Shell or DNS application configuration for specific situation	2–8 wh
Report development (IST Direpo)	8–24 wh
Domain specific application development	8–24 wh

[a]Measured in numbers of average work hours (wh)

half (in total 37%) of the business functionality was provided without programming, i.e. by defining and configuring the business model and IST DNS applications.

The experience of use of the ISTechnology testifies the following development effort (Table 2):

From Table 2 it can be concluded that effort decreases three to four times, if the development activity is to be accomplished by changes in the organization business model or by configuring IST DNS applications. Taking into consideration the number of applications of various categories in the abovementioned IS, it can be concluded that development effort of these IS, in comparison with the situation, if all applications should be developed by programming, has decreased approximately 4 times.

4.3 Maintenance Effort Analysis

More than 600 change requests (CR) had been registered since 2004. The results of the change requests analysis are depicted in Fig. 5.

According to the experience of use of the ISTechnology, if in the IS maintenance process the change request is implemented by changing only the platform independent business model or IST DNS applications, the economy of effort is approximately

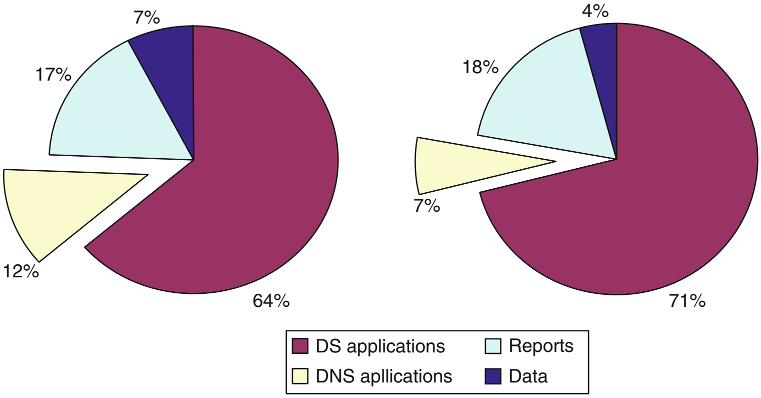


Fig. 5 The percentage of the total number of CR on the left and the percentage of consumed time on CR on the right

two times (see Fig. 5). However, as it might be seen, such change requests make approximately 12% of the total number of change requests. On the one hand it testifies, that the business modeling language is understandable enough for customers and enables to define this model quite precisely already at the moment of development. On the other hand it means that it is essential to widen the range of IST DNS applications and business modeling facilities.

5 Conclusions and Future Work

This paper presents integrated approach to the development of the organization business model and appropriate IS. The experience obtained in practice testifies, that the ISTechnology meta-model can be successfully used in defining of the organization business model. It is easily perceptible to the users and enables to involve users actively in definition, optimization and maintenance of the platform independent business model. The platform specific models are not described by meta-models, but there were ISTechnology applications for specific platforms developed. In that way, the business model is supported with appropriate IS functionality in specific platform.

The experience of the use of the ISTechnology shows, that approximately 40% of the IS functionality and 12% of the IS change requests are implementable by performing changes in the platform independent business model and in domain nonspecific applications. Taking into account, that effort of implementation of new functions and change requests in the business model and in domain nonspecific components are two to four times less effort-consuming, it can be concluded that the ISTechnology enables to economize considerably the effort and cost of IS development and maintenance.

The experience of use of the ISTechnology has outlined also the following most essential directions of development:

- Integration of various organization business models. Already now ISTechnology is used in situations, where, for example, one bank and its branch enterprises have several IS, which are supporting integrated business processes.
- Improvement of the meta-model, by providing more detailed definition of the business goals of organizations (answering the question "Why?").
- Wider provision of IS functionality with domain nonspecific components. The analysis of the developed till now applications shows, that their number, in comparison with the existing one, can be increased by 20% and accordingly the development and maintenance effort of IS can be decreased significantly.

Acknowledgements This research was partially funded by the ESF project "Scientific staff for ESF project implementation". ESF project No. 2009/0216/1DP/1.1.1.2.0/09/APIA/VIAA/044.

References

1. Treimanis M (1998) ISTechnology—technology based approach to information system development. In: Proceedings of the 3rd international baltic workshop "databases and information systems", vol 2. Riga, Latvia, pp 76–90
2. Iljins J (2004) Design interpretation principles in development and usage of informative systems. In: Scientific papers, vol 669. University of Latvia, Latvia, pp 99–106
3. Iljins J (2008) Metamodel based approach to IS development and lessons learned. In: Databases and information systems Baltic DB&IS, pp 167–178
4. Patrascoiu O (2005) Model driven language engineering, a thesis submitted to the University of Kent for the degree of Doctor of Philosophy
5. Kalnins A (2002) Business modelling languages and tools. Progress in industrial mathematics at ECMI. Springer, Heidelberg
6. OMG Unified Modeling LanguageTM (OMG UML) (2009) Superstructure, version 2.2. http://www.omg.org/spec/UML/2.2/Superstructure
7. Rumbaugh J et al (1991) Object-oriented modeling and design. Prentice-Hall, Englewood Cliffs
8. Quatrani T (1999) Visual modeling with rational rose 2000 and UML. Addison Wesley, Upper Saddle River. ISBN: 0-201-69961-3
9. Kalnins A, Barzdins J, Kalis A (1996) GRADE-BM: modeling and simulation facilities. In: Proceedings of the NWPER'96. Aalborg University, Aalborg, pp 71–86
10. Intalio|Cloud. http://www.intalio.com/
11. Web Services Business Process Execution Language Version 2.0 (2006) Public review draft. http://docs.oasis-open.org/wsbpel/2.0/wsbpel-specification-draft.html
12. Knöll HD, Kuhl LWH, Kuhl RWA, Moreton R (2001) Optimising business performance with standard software, Birkhäuser. ISBN 3528057653, 783528057657
13. Dorner C, Draxler S, Pipek V, Wulf V (2009) End users at the bazaar: designing next-generation enterprise resource planning systems. IEEE Software 26(5):45–51
14. Software Product Lines. http://www.softwareproductlines.com/
15. White J, Hill JH, Tambe S, Gokhale AS, Schmidt DC (2009) Improving domain-specific language reuse with software product line techniques. IEEE Software Domain-Specific Model 26(4):47–53
16. Clements P, Northrop L (2002) Software product lines: practices and patterns. Addison Wesley, Upper Saddle River
17. Kleppe A, Warmer J, Bast W (2003) MDA explained: the model driven architecture™: practice and promise. Addison Wesley, Upper Saddle River. ISBN: 0-321-19442-X

18. MetaEdit+. http://www.metacase.com/
19. Sowa JF, Zachman JA (1992) Extending and formalizing the framework for information systems architecture—technical. IBM Syst J 31(3):590
20. Nalbone J, Ambler SW, Vizdos MJ (2005) The enterprise business modeling discipline. Prentice-Hall, Upper Saddle River

Assessment of Ontology Development and Management Tools According to the Axioms

Diana Kalibatiene and Olegas Vasilecas

Abstract Nowadays, there are more than 50 different ontology development and management tools (ODMT), since domain ontologies become used more and more often in the development of knowledge based information systems (IS). A number of authors propose their criteria to assess ODMT. However, the earlier proposed criteria of ODMT assessment concentrate on the modelling capabilities of the structure of an ontology and user interfaces mainly. In this research, we are interested in the ODMT capability of modelling axioms. Therefore, this paper presents an investigation of ODMT according to the modelling of axioms. We define necessary criteria for the assessment of ODMT and perform an evaluator case study.

1 Introduction

Nowadays, there are more than 50 different ontology development and management tools (ODMT), since domain ontologies become used more and more often in the development of knowledge based information systems (IS). A number of authors, like Lavbic and Krisper (2010), propose their methodologies for creating ontologies and authors, like Su and Ilebrekke (2002), Corcho et al. (2003), and Casely-Hayford (2005), proposes their criteria to choose a suitable ODMT. However, the earlier proposed criteria of ODMT assessment concentrate on the modelling capabilities of the structure of an ontology and user interfaces mainly. Modelling capabilities of axioms are mentioned only. In this research, we aim to fill this gap. Therefore, this paper presents an investigation of ODMT according to the modelling of axioms.

The paper is structured as follows. First section presents the related work on decision making. Second section presents selecting criteria and defining a formula for the assessment of ODMT. Third section presents ODMT, which are compared in this research. Fourth section describes the assessment of chosen ODMT. Finally, fifth section concludes the paper.

D. Kalibatiene (✉)
Vilnius Gediminas Technical University, Sauletekio al. 11, 10223, Vilnius, Lithuania
e-mail: diana@isl.vgtu.lt

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_26,

2 Related Work

The problem of measuring according to number of criteria is known as multi-criteria decision making (MCDM). According to a number of authors (Triantaphyllou 2000; Zimmermann 1996), MCDM is divided to multi-objective decision making (MODM) and multi-attribute decision making (MADM). However, very often the terms MADM and MCDM are used to mean the same class of model, i.e. MCDM. MODM studies decision problems in which the decision space is *continuous*. On the other hand, MADM studies problems with *discrete* decision space. However, many of MCDM methods have certain aspects in common (Triantaphyllou 2000). They are: alternatives and attributes.

Alternatives represent different choices available to the decision maker. Attributes (referred to as "*goals*" or "*decision criteria*") represent the different dimensions from which the alternatives can be viewed. Since different criteria represent different dimensions of the alternatives, they may *conflict* with each other. For instance, cost may conflict with profit. Different criteria may be associated with different units of measure. For instance, cost and distance may be measured in terms of litas and kilometre. Therefore, some authors, like Zavadskas (1987), propose to normalise criteria. In this sense, *normalisation*—is the process, during which criteria with different dimensions translated into criteria without dimensions.

It is accepted that an MCDM problem is expressed in a matrix format. A decision making matrix A is an $(m \times n)$ matrix in which element a_{ij} is the performance of alternative A_i when it is evaluated in terms of decision criterion C_j (for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$) (Triantaphyllou 2000). It is also assumed that the decision maker has determined the weights of relative performance of the decision criteria. Weights are denoted as w_j for $j = 1, 2, \ldots, n$. Formulas used for normalization of decision matrices depend on MCDM method used for decision making.

The main steps in utilizing any decision-making technique are as follows: (1) Determine the relevant criteria and alternatives; (2) Attach numerical measures to the relative importance of the criteria and to the impacts of the alternatives on these criteria; (3) Process the numerical values to determine a ranking of each alternative.

Nowadays, there are a number of different MCDM methods. Some of them are presented in Triantaphyllou (2000). However, there is no the best method. In our research we use the WSM (*the weighted sum model*) method, probably the most commonly used method. According to this method, if there are m alternatives and n criteria then the best alternative is the one that satisfies the following expression (formula 1):

$$A^{*}_{WSM-score} = \max_i \sum_{j=1}^{n} a_{ij} w_{ij}, \quad \text{for} \quad j = 1, 2, \ldots, n \tag{1}$$

where: $A^{*}_{WSM-score}$ is the WSM score of the best alternative, n is the number of decision criteria, a_{ij} is the actual value of the i-th alternative in terms of the j-th criterion, and w_j is the weight of importance of the j-th criterion.

3 Selecting Criteria for the Assessment of ODMT

Before selecting criteria for the assessment of ODMT, the term of ontology should be defined. Nowadays, there are a lot of definitions of ontology. Classical definition, borrowed from philosophy, says that ontology means a systematic account of Existence. In computer science, Gruber defines ontology, as a specification of a conceptualisation (Gruber 1995). According to Genesereth and Nilsson (1987), a conceptualisation includes the objects and their relations, which an agent presumes to exist in the world. The process of a conceptualisation is the process of mapping an object or a relation in the world to a representation in our mind.

According to Guarino (1998), "an ontology is a logical theory accounting for the intended meaning of a formal vocabulary, i.e. its ontological commitment to a particular conceptualisation of the world".

However, the modern definition of ontology is extended in term of instances (in Protégé[1] ontologies) or individuals (in Ontology Web Language (OWL) (OMG 2005) based ontologies), which represent objects in the domain that we are interested in (Also known as the *domain of discourse*).

According to the content of a business domain knowledge, ontologies can be: *lightweight*, which describes a hierarchy of concepts related by particular relationships (e.g. is-a, part-of, etc.); *light heavyweight*, in which constraints are added to restrict the values of concepts and relationships, like cardinality constraints, etc.; and *heavyweight*, in which suitable axioms are added in order to express and restrict complex relationships between concepts and to constrain their intended interpretation.

The most existing ontologies, like WordNet,[2] Protégé[3] ontologies (not all), ontologies presented by Culmone et al. (2002) and Lin et al. (2001), DBpedia,[4] are lightweight or light heavyweight, since those have no axioms. In heavyweight ontologies axioms defined in a framework of a description logics (McGuinness and Patel-Schneider 1998), like KIF (Genesereth 2006) in Protégé ontology (Noy et al. 2000) and SUMO[5].

The following definition of ontology is used here, since the concept of an axiom is important in this research. Ontology defines the basic concepts, their definitions and relationships comprising the vocabulary of an application domain and the axioms for constraining interpretation of concepts and expressing complex relationships between concepts (Vasilecas et al. 2009). Some authors, like Falbo et al. (1998), distinguish properties from concepts also.

---

[1] http://protege.stanford.edu.

[2] http://wordnet.princeton.edu/.

[3] http://protege.stanford.edu.

[4] http://dbpedia.org/.

[5] http://www.ontologyportal.org/.

In this research we aim to compare ODMT according to axioms and their modelling possibilities. Therefore, we select the following criteria, according to which existing ODMT will be analysed. They are:

1. ODMT should support modelling of axioms:
   a. ODMT should support an axiom definition language.
   b. ODMT should support an axiom management language.
   c. ODMT should support syntactical checking of axioms.
   d. ODMT should have possibility to process axioms and display results of processing.
2. ODMT availability—ODMT should allow the free open source software, which can be installed locally.
3. ODMT usage:
   a. ODMT should by user-friendly.
   b. ODMT should support graphical notation.
   c. ODMT software should be supported by active project.
4. ODMT should be extensible.

The possible values and weights of the defined criteria are presented in Table 1. A weight of the criterion equals to the importance of this criterion divided to the sum of importance of all criteria. Importance is a whole number from 1 to 9, where 9 is the most important and 1 is the least important. Moreover, a number can appear only once evaluating an importance.

Since the possibility of definition and management of axioms is the most important in this evaluation, we attach the following importance to criteria: 1.1 equals to 9, 1.2—8, 1.3—6, 1.4—2, 2—7, 3.1—1, 3.2—3, 3.3—4, and 4—5. Weights are calculated and presented in Table 1.

4 ODMT Compared in this Research

For the detailed study we choose the most popular *WebODE* (Gómez-Pérez et al. 2003), *OilEd* (Bechhofer 2000), *Ontolingua* (Ontolingua 2005), *Protégé* (Noy et al. 2000), *Chimera* (McGuinness et al. 2000), *OntoSaurus* (Swartout et al. 1997), *OntoEdit* (Sure et al. 2002) and *WebOnto* (Domingue 1997) tools.

The ***Ontolingua*** server was the first ontology tool developed in the knowledge systems laboratory (KSL) at Stanford University in the early 1990s (Farquhar et al. 1997). It was built to ease the development of Ontolingua ontologies. Initially the main module inside the ontology server was the ontology editor and other modules, like Webster (an equation solver) and OKBC (Open Knowledge Based Connectivity) server.

OntoSaurus was developed around the same time as Ontolingua by the Information Sciences Institute (ISI) at the University of South California (Swartout et al. 1997). OntoSaurus consists of two modules: an ontology server, which uses LOOM

Table 1 Vales and weights of the selected criteria

Criteria (C_i)	Possible values	Weights (w_j)
Axiom definition language (1.1)	1—ODMT supports an axiom definition language; 0—ODMT doesn't support an axiom definition language;	0,2
Axiom management language (1.2)	1—ODMT supports an axiom management language; 0—ODMT doesn't support an axiom management language;	0,1778
Syntactical checking of axioms (1.3)	1—ODMT supports axiom checking; 0—ODMT doesn't support axiom checking;	0,1333
Processing of axioms, inference (1.4)	1—ODMT supports axiom processing; 0—ODMT doesn't support axiom processing;	0,0444
Availability (2)	1—a free open source software, which can be installed locally, is available; 0.75—Web access to the free version of a software is available 0.5—a free evaluation version, which can be installed locally, is available; 0.25—free Web access to evaluation version; 0—a software licence is necessary;	0,1556
User-friendly (3.1)	1—yes; 0—no;	0,0222
Graphical notation (3.2)	1—graphical taxonomy, graphical view; 0.5—graphical taxonomy, no graphical view; 0—no;	0,0667
Active project (3.3)	1—yes; 0—no;	0,0889
Extensible (4)	1—yes (plug-ins) 0—no	0,1111
	Total value	*1*

as its knowledge representation system, and a web browser for LOOM ontologies. There are translators from LOOM to Ontolingua, KIF, KRSS (Knowledge Representation System Specification) and C++. OntoSaurus ontologies can also be accessed with the OKBC protocol.

WebOnto is an ontology editor for OCML (Operational Conceptual Modelling Language) ontologies and was developed at the Knowledge Media Institute (KMI) at Open University (Domingue 1998). This tool is a Java applet coupled with a customised web server and allows users to browse and edit knowledge models over the internet. The fact that WebOnto was able to support collaborative ontology editing was a major advantage at the time.

Ontolingua, Ontosaurus and WebOnto were created solely for the purpose of browsing and editing ontologies in a specific language (Ontolingua, LOOM and OCML respectively). These older generation editors were hardly extensible compared to the engineering environments of today. The newer generation of ontology-engineering environments are more advanced and ambitious than their predecessors. They are extensible, have component based architectures, where new modules can easily be added to provide more functionality to the environment.

WebODE is an easily extensible and scalable ontology workbench developed by the Ontology Group at the Technical University of Madrid (UPM) (Arpiirez et al. 2001). It is the successor of the Ontology Design Environment (ODE). WebODE is used as a Web server with a Web Interface. The core of this environment is the ontology access service which is used by all the services and applications plugged into the server. The ontology editor also provides constraint checking capabilities, axiom rule creation and parsing with the WebODE Axiom Builder (WAB) editor, documentation in HTML (Hyper text Markup Language), an ontology merge, and ontology exportation and importation in different formats (XML\RDF(s), OIL, DAML+OIL, CARIN, F logic, Java and Jess). Its inference built in service uses Prolog and a subset of the OKBC protocol.

OilED is an ontology editor that allows the user to build ontologies in OIL (the Ontology Inference Layer) and DAML+OIL (DARPA (Defense Advanced Research Projects Agency) Agent Markup Language) (Bechhofer et al. 2001). OilED is a DL (Description Logics) based tool and is installed locally. OilED was developed by The University of Manchester, the Free University of Amsterdam and Enterprise GmbH. The current versions of OilED do not support the development of large scale ontologies, in terms of migration, integration, versioning argumentation and other activities involved in ontology construction.

OntoEdit is developed by AIFB, University of Karlsruher and is built on top of a powerful internal ontology model (Sure et al. 2002). The internal ontology model can be serialised using XML, which supports the internal file handling. It supports F-Logic, RDF-Schema and OIL. In the current version OntoEdit has an interface to the Karlsruher F-Logic Inference Engine (the backbone of OntoBroker). It is stated that in the next version the FaCT system will be accessible from OntoEdit. The tool is based on a flexible plug-in framework. The professional version of OntoEdit contains several additional plug-ins, a collaborative environment and inference capabilities.

Chimera is developed in the Knowledge based systems Laboratory at Stanford University (McGuinness et al. 2000). It was built on top of Ontolingua and is a software system that supports users in creating and maintaining distributed ontologies on the web. The user interacts with Chimera through a browser such as Netscape or Microsoft Internet explorer. It supports users in tasks like loading knowledge bases in differing formats, reorganizing taxonomies, resolving name conflicts, browsing ontologies and editing terms.

Protégé is one of the most widely used editing tools and has been developed by the Stanford Medical Informatics (SMI) at Stanford University (Noy et al. 2000). The design and development of Protégé has been driven primarily by two goals: to be compatible with other systems for knowledge representation and to be an easy to use and configurable tool for knowledge extraction. It is an open source, standalone application with an extensible architecture, which assists users in the construction of large electronic knowledge bases. The core of this environment is an ontology editor. Numerous plug-ins provide several functions including alternative visualization mechanisms, management of multiple ontologies, inference services and ontology language importation/exportation.

Table 2 Assessment of *WebODE*, *OilEd*, *Ontolingua*, *Protégé*, *Chimera*, *OntoSaurus*, *OntoEdit* and *WebOnto* ontology development and management tools

Criteria \ ODTM	WebODE	OilEd	Ontolingua	Protégé	Chimera	OntoSaurus	OntoEdit	WebOnto
Axiom definition language (1.1)	Yes (WAB)	Yes (DAML + OIL)	Yes (KIF)	Yes (PAL, SWRL)	Yes (KIF)	Yes (KIF)	Yes (F Logic)	Yes (OCML)
Axiom management language (1.2)	Yes (WAB)	Yes (DAML + OIL)	Yes (KIF)	Yes (PAL, SWRL)	Yes (KIF)	Yes (KIF)	Yes (F Logic)	Yes (OCML)
Checking of axioms (1.3)	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes
Processing of axioms (1.4)	No	Yes	No	Yes	Yes	Yes	Inference	Yes
Availability (2)	Free Web Access	6	Free Web Access	6	7	7	8	Free Web Access
User-friendly (3.1)	Yes	No	Yes	Yes	Yes	No	No	Yes
Graphical notation (3.2)	9	No	10	9	9	No	No	9
Active project (3.3)	No	No	Yes	Yes	Yes	Yes	Yes	No
Extensible (4)	No	No	No	Plug-ins	Plug-ins	No	Plug-ins	No

[6]Open source, installed locally
[7]Open source and free Web access to evaluation version
[8]Free web access to free version. OntoEdit Professional needs software licence
[9]Graphical taxonomy, graphical view
[10]Graphical taxonomy, no graphical view

Table 3 Primary decision-making matrix for the assessment of ODMT

	ODTM							
Criteria	**WebODE**	**OilEd**	**Ontolingua**	**Protégé**	**Chimera**	**OntoSau-rus**	**OntoEdit**	**WebOnto**
Axiom definition language (1.1)	1	1	1	1	1	1	1	1
Axiom management language (1.2)	1	1	1	1	1	1	1	1
Checking of axioms (1.3)	1	1	0	1	1	1	1	1
Processing of axioms (1.4)	0	1	0	1	1	1	0	1
Availability (2)	0,75	1	0,75	1	0,5	0,5	0,25	0,75
User-friendly (3.1)	1	0	1	1	1	0	0	1
Graphical notation (3.2)	1	0	0,5	1	1	0	0	1
Active project (3.3)	0	0	1	1	1	1	1	0
Extensible (4)	0	0	0	1	1	0	1	0
$A_{WSM\text{-}score}$	0,7167 7	0,7111 1	0,63895	1	0,9222 2	0,7222 2	0,75	0,7611 1

5 Assessment of Chosen ODMT

The results of the ODML assessment are presented in Table 2, which describes each ODMT according to the defined criteria in details.

At the next step, we transform Table 2, which describes each criterion by text, into Table 3, which presents numerical values of criteria. Table 2 is used in this step.

$A_{WSM\text{-}score}$ is calculated according to the formula 1 and holds the WSM scores of alternatives. The Protégé ODMT has the best evaluation according to the defined criteria among chosen ODMT.

6 Conclusions

According to the defined criteria in terms of axiom modelling capability, eight ontology development and management tools (*WebODE*, *OilEd*, *Ontolingua*, *Protégé*, *Chimera*, *OntoSaurus*, *OntoEdit* and *WebOnto*) are compared, and the Protégé is chosen as the most suitable. However, Chimera stands behind according to availability, since the full version requires a licence.

The proposed approach of assessment can be extended in terms of criteria and new alternatives, if necessary. Moreover, it is universal, e.g. it can be applied for assessing alternatives in different areas.

In the future work we are going to refresh the list of analysed ODMT and assess languages, used to define ontologies with axioms.

References

Arpiirez JC, Corcho O, Fernaandez-Loopez M, Goomez-Peerez A (2001) WebODE: a scalable ontological engineering workbench. In Gil Y, Musen M, Shavlik J (eds) First international conference on knowledge capture. ACM Press, New York, pp 6–13

Bechhofer S (2000) OilEd ontology editor. University of Manchester. http://xml.coverpages.org/oilEdANn20001204.html. Accessed June 2008

Bechhofer S, Horrocks I, Goble C, Stevens R (2001) OilEd: a reasonable ontology editor for the semantic web. In: Baader F, Brewka G, Eiter T (eds) Proceedings of the joint German/Austrian conference on artificial intelligence (KIÕ01), LNAI 2174. Springer, London, pp 396–408

Casely-Hayford L (2005) A comparative analysis of methodologies, tools and languages used for building ontologies. CCLRC Daresbury Laboratories. http://epubs.cclrc.ac.uk/bitstream/894/OntologyReport.pdf. Accessed June 2008

Corcho O, Fernandez-Lopez M, Gomez-Perez A (2003) Methodologies, tools and languages for building ontologies. Where is their meeting point? Data Knowl Eng 46(1):41–64

Culmone R, Rossi G, Merelli E (2002) An ontology similarity algorithm for BioAgent. In: Ercolani S, Zamboni MA (eds). NETTAB workshop on agents and bioinformtics. http://www.bioagent.net/WWWPublications/Download/NETTAB02P1.pdf. Accessed Mar 2007

Domingue J (1998) Tadzebao and Webonto: discussing, browsing and editing ontologies on the web. In: Proceedings of the 11th knowledge acquisition workshop (KAW98). http://ksi.cpsc.ucalgary.ca/KAW/KAW98/domingue/. Accessed Dec 2008

Domingue J (2008) WebOnto. http://kmi.open.ac.uk/projects/webonto/. Accessed June 2008

Falbo RA, Menezes CS, Rocha ARC (1998) A systematic approach for building ontologies. In: Coelho H (ed) Proceedings of the 6th Ibero-American conference on AI (IBERAMIA'98), LNAI 1484. Springer, Berlin, pp 349–360

Farquhar A, Fikes R, Rice J (1997) The Ontolingua server: a tool for collaborative ontology construction. Int J Hum Comput Stud 46(6):707–727

Genesereth MR (2006) Knowledge interchange format (KIF). http://logic.stanford.edu/kif/kif.html. Accessed Oct 2006

Genesereth MR, Nilsson NJ (1987) Logical foundations of artificial intelligence. Morgan Kaufmann Publishers, San Francisco

Gómez-Pérez A, Fernàndez-López M, Corcho O (2003) WebODE ontology engineering platform. WebODE Development Group. http://webode.dia.fi.upm.es/WebODEWeb/index.html. Accessed June 2008

Gruber TR (1995) Toward principles for the design of ontolgies for knowledge sharing. Int J Hum Comput Stud 43(4–5):907–928

Guarino N (1998) Formal ontology and information systems. In: Proceedings of FOIS'98. IOS Press, Amsterdam, pp 3–15

Lavbic D, Krisper M (2010) Facilitating ontology development with continuous evaluation. Informatica 21:533–552

Lin S et al (2001) Integrating a heterogeneous distributed data environment with a database specific ontology. In: Sha EHM (ed) Proceedings of the international conference on parallel and distributed systems 2001 (ICPADS'01). IEEE Computer Society Press, Washington, pp 430–435

McGuinness D, Patel-Schneider P (1998) Usability issues in knowledge representation systems. In: Mostow J, Rich C (eds) Proceedings of AAAI-98, Madison, Wisconsin. American Association for Artificial Intelligence, Menlo Park, pp 608–614

McGuinness D et al (2000) The chimaera ontology environment. In: Proceedings of the 17th national conference on artificial intelligence and 12th conference on innovative applications of artificial intelligence. AAAI Press/The MIT Press, pp 1123–1124

Noy NF, Fergerson RW, Musen MA (2000) The knowledge model of protege-2000: combining interoperability and flexibility. In: Dieng R, Corby O (eds) Proceedings of the 12th international conference in knowledge engineering and knowledge management (EKAWÕ00), LNAI 1937. Springer, Berlin, pp 17–32

OMG (2005) Ontology definition metamodel. http://www.omg.org/docs/ad/05-08-01.pdf. Accessed Sept 2008

Ontolingua (2005) Ontolingua—software description. Stanford University. http://www.ksl.stanford.edu/software/ontolingua/. Accessed June 2008

Su X, Ilebrekke L (2002) A comparative study of ontology languages and tools. In: Pidduck AB et al (eds) Proceedings of the 14th international conference (CaiSE), LNCS 2348. Springer, London, pp 761–765

Sure Y et al (2002) OntoEdit: collaborative ontology engineering for the semantic web. In: First international semantic web conference (ISWCÕ02), LNCS 2342. Springer, Berlin, pp 221–235

Swartout B et al (1997) Toward distributed use of large-scale ontologies. In: Gaines B, Musen M (eds) Proceedings of the 10th knowledge acquisition for knowledge-based systems workshop (KAW'96). http://ksi.cpsc.ucalgary.ca/KAW/KAW96/swartout/Banff_96_final_2.html. Accessed Apr 2009

Triantaphyllou E (2000) Multi-criteria decision making methods: a comparative study. Springer, New York

Vasilecas O, Kalibatiene D, Guizzardi G (2009) Towards a formal method for transforming ontology axioms to application domain rules. Inf Technol Control 38(4):271–282

Zimmermann HJ (1996) Fuzzy set theory and its applications, 3rd edn. Kluwer, Boston

Завадскас ЕК (1987) *Комплексная оценка и выбор ресурсосберегающих решений в строительстве*. Мокслас, Вилнюс

Semi-Automatic Transformation of an XML Schema to XForms

Ján Kasarda and Tomáš Bartoš

Abstract The capabilities of classic web forms written purely in HTML become these days insufficient. The missing type control, no support for validation, or tight coupling of data and presentation layers limit their usage. So the *XForms* technology, the new generation forms based completely on the well-known XML format, was created. It includes all mentioned features and no other scripting language is needed. There are several studies of how to automate the process of creating XForms if there exists a schema against which the form will be validated. This paper presents a new method of *semi-automatic transformation* from an *XML Schema* to *XForms* using a simple subset of the XHTML as the host language for generated forms. The proposed algorithm is based on traversing the input XML schema and generating the XForms form in two phases. We prove the feasibility of this concept with the implemented XForms editor.

1 Introduction

Today, more and more web sites require users to register and login to view the content, to fill some form to proceed with online shopping, or to agree with license terms before downloading a file from a server. In all these situations, the web forms are used to handle the input data. The most popular and widely used technology for creating diverse forms on web sites is the HTML [12] or the XHTML [14] language. However, these technologies allow creating web forms with only a simple functionality where users are able just to type the text or select a value. They were not designed to support the dynamic behavior such as the validation or the type control of input controls. Therefore, we need to use other tools such as the JavaScript [2] language to enable advanced features.

J. Kasarda (✉)
Department of Software Engineering, Malostranské náměstí 25, 118 00, Prague 1, Czech Republic
e-mail: jan.kasarda@mff.cuni.cz

T. Bartoš
e-mail: bartos@ksi.mff.cuni.cz

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_27,

The alternative to the classic web forms is a new W3C technology called *XForms* [17]. XForms are the second generation forms that separate the presentation layer from the data layer. Furthermore, the data layer is built completely on top of the XML format, so the data from an XForms form is represented as a well-formed XML document with a predefined structure. We describe the structure of the XML documents with various languages such as XML Schema [15] or DTD [8]. XForms use XML Schema as the structure description language.

These documents might be processed further by other technologies such as XSLT transformations [18] or they can be used as input parameters for a web service. Because we know the structure of the documents, we can validate the forms before sending to a server or other recipients.

A common situation is that we have an XML schema describing XML documents that are to be edited by users through a web form. Therefore, it is natural to help a form designer to derive the form from the XML schema. More precisely, it means to derive an XForms code describing the form from the supplied XML schema. The ability of generating forms from an XML schema brings better flexibility when we create applications specific for a given set of XML schemas. It also makes easier the maintenance of big collections of XML schemas and related forms.

Therefore we decided to implement an XForms editor [5], [6] that

- helps designers to transform the given schema that describes the structure of XML documents to the XForms form
- takes XML Schema as the language for the structural description
- includes algorithms that provide semi-automatic transformation of XSDs (XML Schema Definitions) to XForms forms in two phases (see Sect. 4)

2 Related Work

Generating an XForms form from a given XML schema is not a new concept. There exists several transformation methods that we can categorize according to two basic criteria: the *degree of automation* and the *implementation technique.*

2.1 Degree of Automation

The degree of automation divides the approaches into two groups:

- **Automatic Transformations.** These solutions (e.g. [3, 8]) generate the XForms form automatically from the given XML schema with no user interaction. This is useful when the XML schema is quite large, complex, or when there is a big number of XML schemas. The disadvantage is that the form designer cannot influence the transformation process and we must provide a default solution for the case of ambiguity. We also have to manually add further information to the generated XForms form, e.g. the visual layout of the form or labels for its controls.

- **Semi-Automatic Transformations.** The process is similar to automatic transformation but the main advantage is that, in case of ambiguity, we can ask the form designer how to solve the situation. So he or she has a chance to influence the final XForms form and also directly modify the form layout and all form controls during the transformation process. A famous implementation of the semi-automatic transformations is the *Visual XForms Designer* [4].

2.2 Implementation Techniques

On the other hand, the implementation technique means how the solutions access and handle the XML schema languages. The known methods use:

- **XML-based Access.** These concepts leverage the fact that both, the XML schema languages (except for the DTD) and XForms, are based on the XML format. So the implementations provide just an XSLT script that transforms the given XML schema to the code of the corresponding XForms form. While the advantage is the effectiveness of such implementations, the drawback is that the XSLT is not a programming language, so it cannot interact with the form designer. Therefore we can use XSLT only for automatic methods.
- **Object-Oriented Technique.** We can use object-oriented languages based on usual XML parsing technologies, e.g. DOM [13] or SAX [1]. They provide user-friendly API and allow the interaction with the form designer. But we must know the language and typically we need more time to implement the solution.

2.3 Existing Solutions

The interesting work [8] presents an automatic method of generating XForms forms from the DTD. Authors use XSL [16] in the XML document to implement certain functionality. The proposed method assumes some restrictions on the input DTD.

The generic XSD-to-XForms processor [3] is also a representative of the strictly automatic translation implemented on top of the XSLT which takes an XML schema expressed in XSD as the input. Generally, it is a very advanced approach that considers several possible cases including different naming conventions or imported namespaces.

More business-oriented implementation is under the development of the company IBM. As part of their AlphaWork project [4] they developed the *XML Forms Generator,* a plug-in for Eclipse IDE. This tool represents a semi-automatic transformation and it generates XForms forms from the XML Schema. Users can create a form from the part of the schema or there is an option to create the form using WYSIWYG (What You See Is What You Get) editor that enables adding input elements to schema using the drag-and-drop method.

Thanks to the separation of the data from the presentation layer, we are able to store data in XForms in XML format. This feature makes XForms to be a perfect candidate for creating Web services interfaces which was elaborated in [9]. Experimental results of XForms communicating with Web services are very impressive.

Considering the existing solutions, we decided to use a procedural language (leveraging the object-oriented concepts) to create a new algorithm for the semi-automatic transformations. This combination is user friendly and we can easily enhance the performance of this approach in the future.

3 XML Technologies

Extensible Markup Language or XML was developed primarily as the format for storing the semi-structured data. It became very popular and it proved to be useful also in several different areas such as data modeling (XML Schema) or data transformations (XSLT [18]).

3.1 XML Schema

The XML Schema is one of the languages that define a structure of XML documents and it is completely based on the XML format; see the sample XML Schema in Fig. 1a. An XML Schema defines several conditions that the XML documents must meet to be valid against this schema. Such conditions might be the specific value types or the predefined structure of documents. Additionally, we can define custom data types or just use the built-in simple types.

The language was developed by the W3C consortium and we use it as the structure description language for our algorithm.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 targetNamespace="http://xforms-builder.com"
 xmlns="http://xforms-builder.com">
 <xs:element name="contactList">
  <xs:complexType>
   <xs:sequence>
    <xs:element name="contact"
                minOccurs="1"
         maxOccurs="100">
     <xs:complexType>
      <xs:sequence>
       <xs:element name="name"
                   type="xs:string"/>
        <xs:choice>
         <xs:element name="phone"
                     type="xs:string"/>
         <xs:element name="email"
                     type="xs:string"/>
        </xs:choice>
       </xs:sequence>
      </xs:complexType>
     </xs:element>
    </xs:sequence>
   </xs:complexType>
  </xs:element>
</xs:schema>
```

a

```
<xf:model id="Model" schema="schema.xsd">
 <xf:instance id="contactList">
  <contactList>
    <contact temp="phone">
    <name />
    <phone />
    <email />
   </contact>
  </contactList>
 </xf:instance>
 <xf:bind nodeset="contact/name" type="xsd:string"/>
 <xf:bind nodeset="contact/@temp" relevant="false"/>
 <xf:bind nodeset="contact/phone" type="xsd:string"
    relevant="../@temp = 'phone'" />
 <xf:bind nodeset="contact/email" type="xsd:string"
    relevant="../@temp = 'email'" />
 <xf:submission id="id"
  action="http://xformstest.org/cgi-bin/showinstance.sh"
  method="put"/>
</xf:model>

<xf:group ref=".">
 <xf:label>contact</xf:label>
 <xf:group ref="./name">
  <xf:label>name</xf:label>
  <xf:input incremental="true" model="Model" ref="." />
 </xf:group>
  ...
</xf:group>
```

b

Fig. 1 The sample XML files: **a** sample XML schema, **b** sample XForms form

3.2 XPath

When we use semi-structured documents, we want to access specific elements, search for particular attributes, or find certain values. For this purpose, several languages have been created and one of them is the XPath language. We can navigate through elements with absolute and relative paths or define specific parts of the documents using XPath expressions.

In our context, we use XPath expression to define bindings in the XForms (explained in Sect. 4.3).

3.3 XForms

XForms are valid XML documents that are intended to replace the existing web forms in the future. The dynamic behavior, the support for repeating controls or the validation, and the separation of the data layer from the presentation layer (the Model-View-Controller approach) belong to the main advantages.

Even though XForms contain the user interface, we have to supply the host languages into which we insert the XForms form to create a well-formed XML (e.g. HTML or XHTML for viewing in web browsers).

To understand the principles of XForms and the proposed algorithms (Sect. 4), we need to define several terms that we will use in the following text.

Definition 1. XSD Node An XSD node is a 5-tuple $N =$ (name, min, max, nodeType, content) where name is the string identifier of a node, min and max are integers handling minimal and maximal cardinalities of elements in a schema, nodeType $\in$ {element, all, choice, sequence, attribute}, and content is a set of XSD nodes.

Definition 2. XForms Attribute An XForms attribute is a pair $A =$ (name, attrType) where name is the string identifier of an attribute and attrType $\in$ {string, integer}.

Definition 3. XForms Instance Node An XForms instance node is a 4-tuple $I =$ (name, type, children, attrs) where name is the string identifier of a node, type is an XSD node, children stands for an ordered set of XForms instance nodes, and attrs is a set of XForms attributes.

Definition 4. XForms Input Control An XForms input control is a user interface element in XForms. It corresponds to a single Core Form Control Element [17].

Definition 5. XForms Binding An XForms binding is a pair $B =$ (instance, control) where instance is an XForms instance node and control is an XForms input control.

Definition 6. XForms Model An XForms model is a pair $M =$ (instance, bindings) where instance is an XForms instance node and bindings is a set of XForms bindings.

Definition 7. XForms Tree An XForms tree is a pair $T = $ (model, UI) where model is an XForms model and UI is a set of XForms input controls.

Example 1. The sample XForms form (see Fig. 1b) represents an *XForms tree.* The logical structure of the first XML element `xf:model` is divided into two parts, namely *instance* and the *binding* part.

The *instance* is represented by the XML element `xf:instance` and it keeps the data edited by form users while the *binding part* contains several `xf:bind` elements which specify relationships between the data and components of the following *UI* part. Simply speaking, they assign XML schema types to the UI components. This allows the validation of data values filled in the form.

The *UI* specifies the user interface expressed by various XForms elements such as `xf:label` which specifies an input field label or `xf:input` which specifies a single-line text input field.

4 Algorithm

In this section, we introduce a new semi-automatic algorithm that translates the input XML schema expressed in XSD to a corresponding XForms form. The algorithm is able to interact with the form designer who can influence which element and attribute declarations from the XML schema will be reflected in the form and how they will be grouped together, validated, or visualized. The aim is to assist the form designer rather then to do all the work automatically.

The algorithm has two phases. Firstly, it traverses the XML schema and derives the instance part of the resulting form. Secondly, it generates the UI part of the form. It has a built-in behavior which generates some default UI components for each part of the instance. This behavior can however be influenced by the form designer. The binding part is then created automatically.

We will use a sample XML schema depicted in Fig. 1a to generate a corresponding XForms form; see Fig. 1b.

4.1 *Creating the Instance*

We formally specify the first phase of the algorithm in Algorithm 1. We simply traverse the given XML schema from the root to the leaves (top-down approach) and we generate an XML document with XML elements corresponding to the element declarations.

As the product of the first phase we generate an XForms instance node, a "complete" instance with all possible XML elements and attributes that can be derived from the given XML schema. We can extend this instance only by processing the repeating nodes. For our sample XML schema in Fig. 1a, the result is shown in Fig. 2.

Algorithm 1 CREATEINSTANCE(S, D)

Require: XSD source node S; destination XForms instance node D

if S.nodeType = element **then**
  $N \leftarrow$ new XForms instance node
  N.name = S.name
  N.type $\leftarrow S$
  D.addChild(N)
  for all *child* $\in$ *S.content* **do** {for complex types only}
    if *child*.nodeType = attribute **then**
      CREATEINSTANCE(*child*, D)
    else
      CREATEINSTANCE(*child*,N)
    end if
  end for
else if S.nodeType $\in$ {choice, all, sequence} **then**
  for all *child* $\in$ S.content **do**
    CREATEINSTANCE(*child*,D)
  end for
else if S.nodeType = attribute **then**
  $A \leftarrow$ new XForms attribute
  A.name $\leftarrow$ S.name
  D.addAttribute(A)
  D.type $\leftarrow S$
end if

The created instance does not contain any information about types, values, or any other data that we could obtain from the source XML schema. We will append these details in the next phase which derives the UI and binding parts.

4.2 *Creating the XForms Tree*

Algorithm 2 describes the second phase of the algorithm. We traverse the XForms instance created in the first phase from its root to leaves. In each step in the second phase we produce a new binding in the binding part and usually also a new XForms input control component in the UI part of the form using the information from the input XML Schema. Creating the XForms input control is not always necessary.

Fig. 2 Sample XForms instance

```
<contactList>
  <contact temp="phone">
    <name />
    <phone />
    <email />
  </contact>
</contactList>
```

Algorithm 2 CREATEUI(*S*, *D*)

```
Require: source XForms instance node S; destination XForms tree D
parent = new xf:group fdefault parent is a new groupg
if (S:type.min > 1) or (S.type.max > 1) then
    parent = new xf:repeat
end if
D.UI.addControl(parent)
if isSimpleType(S.type) then {attribute or XML Schema simple type}
    E ← createUIElement(S)
    D.UI.addControl(E)
    createBinding(E, S)
else {complex types}
    if S.type.nodeType = choice then
        createPlaceHolder() {updates the XForms instance}
        for all child ∈ S.children do
            createUI(child, parent)
            parent.setRelevancy(child)
        end for
    else
        for all child ∈ S.children do
            createUI(child, parent)
        end for
    end if
end if
```

The way of processing an XForms instance node depends on the type of the corresponding XSD node (the node in the input XML Schema) whether it is a *simple* or a *complex type.* The *simple types* include attributes and XML Schema simple types such as string or integer. All other types are complex.

Now we discuss in more detail the steps of the second phase in terms of all possible cases of processing an XForms instance node and creating the corresponding XForms input control. Then, we explain how we create the bindings.

4.2.1 Simple Types

XForms recommendation [17] created by W3C defines a wide collection of built-in types. For each element of a simple type we create the UI control and bind it to this element. The basic implementations might use the default `<xforms:input>`, advanced solutions might use the suggested UI controls.

4.2.2 Choice NodeType

The basic idea of processing the `<xsd:choice>` element is to create the element `<xf:group>` for each possible choice and let the user choose which part should be applied. We allow the user to choose exactly one option via triggers or another type of input controls.

We store the selected choice in a temporary XML attribute created in the first phase while processing the `<xsd:choice>` element. Then, we assign this attribute to the parent element of all choices (e.g. the `<contact>` element in the sample XForms form). The `createPlaceHolder` () function ensures all these actions and it also generates input controls that we need after a change of the choice.

The temporary attribute is not a part of the XML schema, so we have to mark it as irrelevant. We assign the relevance as an attribute to the corresponding binding element:

```
<xf:bind nodeset="contact/@temp" relevant="false"/>
```

Afterwards, we continue with processing all child elements of the choice by applying the same principles of creating the UI. Because each choice is encapsulated in its own `<xf:group>` element, the function `setRelevancy`() makes these groups visible but only in case that the user choose them. We set the visibility by creating a correct reference to these groups as attributes in the binding element:

```
<xf:bind nodeset="contact/phone" type="xsd:string"
   relevant="../@temp = 'phone'" />
<xf:bind nodeset="contact/email" type="xsd:string"
   relevant="../@temp = 'email'" />
```

We know that `<xf:swich>` element seems to be better alternative for this case but there were some technical issues with processing it in some XForms framework implementations when we implemented this algorithm (e.g. problems with `<xf:switch>` inside `<xf:repeat>` element).

4.2.3 Repeat NodeType

Processing of any `<xsd:element>` with changed cardinality is one of the most complex part of the algorithm. XForms 1.0 do not check minimum and maximum occurrences of a repeating element. It means that XForms repeat placing the element as many times as user wants. Then XForms remove the very last element and as the result, we cannot repeat the action again and add more elements.

In order to prevent XForms from deleting the very last instance of an XSD node with multiple occurrences, we add one extra instance of the XSD node and make it irrelevant to the XML Schema. The form designers do not have to handle any extra item, so there is no change for them. But we must ensure that triggers are aware of this hidden field, so they do not remove it. First, we generate the binding element in the usual way and then we extend it with the *relevant* attribute. The XPath expression only counts the siblings and compares it to zero.

We add the `<xf:repeat>` element to the UI which will change the context for all nested elements that we need to update.

The last thing is that we create the triggers for the add or the delete events. Depending on the choice of users, we might need to generate different XPath expressions in order to address appropriate elements correctly.

4.3 Binding Elements

During the second phase, we also create bindings for each XML node in the XForms instance. The main purpose of a binding is to specify the relationship between an XML node and the corresponding type in the input XML schema. We create these bindings as the `binding` elements.

In any binding element, there must be an XPath expression which addresses the XML node in the XForms instance. In the algorithm, we build the XPath expression by traversing the already generated XForms instance bottom-up, from the current XML node to the root.

Although the context might be changed when adding a new `<xf:repeat>` element, we are able to recover the XPath expressions. Moreover, we might include additional conditions or criteria for the bindings. For example we can add a certain XPath expression to check the number of elements (for repeating elements and their bounds) or to provide a type control.

5 XForms Editor

As a part of the research, we implemented both algorithms and provide a demonstrative plug-in [6] to Eclipse IDE [10] (see Fig. 3). We parse and handle the input XML Schema with Eclipse modeling tools (MDT [11]). We chose XML Schema because of advance capabilities and less limitations when compared to DTD.

Then, we developed the semi-automatic approach with the ability of user interactions via dialog windows (called *wizards*) that are created dynamically according to the current node in the sequence of nodes in the given XML Schema. Wizards help users to choose a correct option in case of ambiguity (mostly for repeat and choice NodeTypes; see Sects. 4.2.2 and 4.2.3) or when users do some customizations.

6 Customization Using CSS

XForms are XML-based technology but in order to use the functionality XForms offer, we have to include the XForms fragments into so called *host language*. This language should be also based on the XML format. The common used host languages are WML for WAP and HTML/XHTML for web sites.

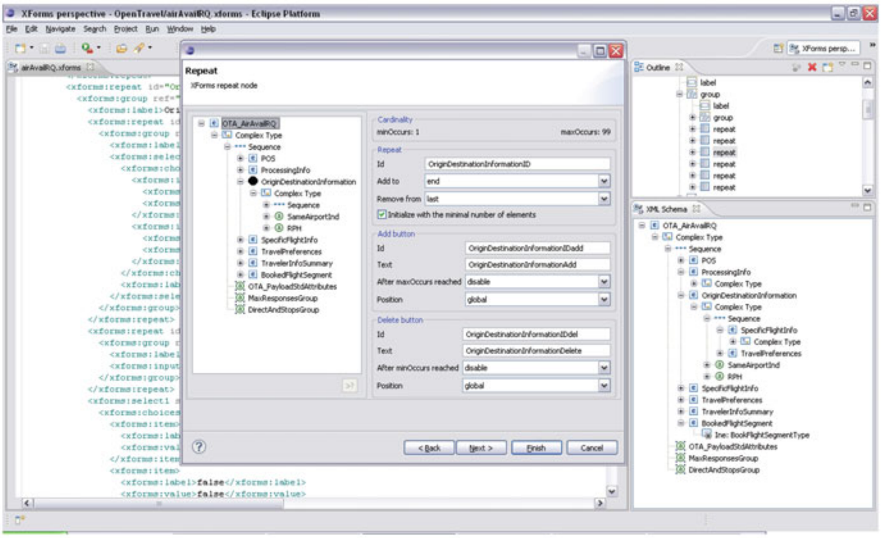
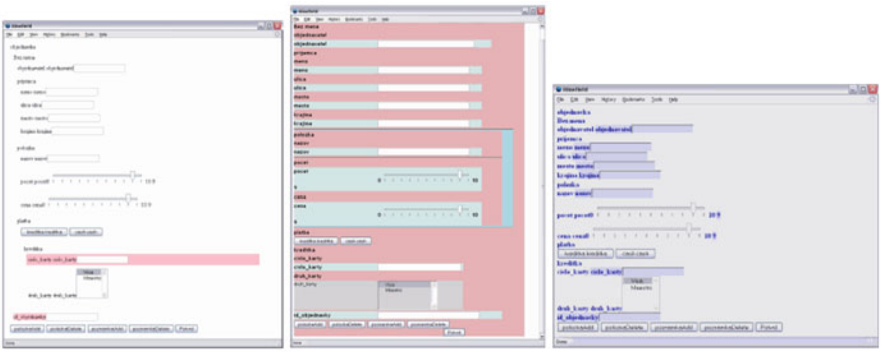


Fig. 3 XForms editor: integration in Eclipse IDE



a b c

Fig. 4 Various CSS customizations of a single XForms form

Furthermore, we can customize XForms that we include in XHTML using CSS (Cascading Style Sheets; see Fig. 4). Typical web forms do not have validating ability and they cannot change their classes dynamically. Because the event model enables validation "on-the-fly", the input controls can change their states from the invalid to the valid state after any user interaction. XForms also introduced some extensions of the class model for input controls, therefore we are able to dynamically change the appearance of input controls using CSS styles.

7 Conclusion and Future Work

The XForms technology showed to be a very useful representation based on the standardized XML format which makes it easy to use. We see the perspective of using XForms in the future, e.g., as a replacement of classic web forms written in HTML. The capabilities of the format such as type control, support for the validation or dynamic behavior, or separation of the data layer using Model-View-Controller make the competitive advantage over other technologies.

But generating XForms forms for a huge number of different documents might be time-consuming, so we focused on the topic of generating XForms forms from a given schema. We took the XML Schema as the input schema that describes the structure of documents, and we proposed a new semi-automatic transformation of the given XML Schema to XForms. The implemented XForms editor might be useful also in the business for easy development of different forms.

We understand that generating XForms is only the first step in the process of managing forms. For the future work, we would like to enhance the transformation, implement the reverse modification of the generated XForms forms, or consider and develop our approach as a part of an XML technology space [7].

Acknowledgements This research has been partly supported by Czech Science Foundation (GACR) project Nr. P202/10/0573.

References

1. Brownell D (2004) SAX. http://www.saxproject.org
2. Flanagan D (2006) JavaScript—the definitive guide: activate your web pages: covers Ajax and DOM scripting, 5. edn. O'Reilly, pp I–XXII, 1–994. http://www.oreilly.de/catalog/jscript5/index.html. (978-0-596-10199-2, DBLP, http://dblp.uni-trier.de)
3. Garvey P, French B (2003) Generating user interfaces from compoite schemas. In: Proceedings of XML 2003, Philadelphia, PA, Dec 2003
4. IBM (2006) Visual XForms designer. http://www.alphaworks.ibm.com/tech/vxd
5. Kasarda J (2007) Editor XForms. Master's thesis, Faculty of Mathematics and Physics, Charles University in Prague
6. Kasarda J (2007) XForms editor. http://www.ms.mff.cuni.cz/ kasaj4am/xforms.zip
7. Kurtev I, Bezivin J, Aksit M (2002) Technological spaces: an initial appraisal. In: International symposium on distributed objects and applications, DOA 2002
8. Lee E, Kim T-H (2005) Automatic generation of XForms code using DTD. In: ICIS'05: proceedings of the 4th annual ACIS international conference on computer and information science. IEEE Computer Society, Washington, DC, pp 210–214
9. Song K, Lee K-H (2007) An automated generation of XForms interfaces for web service. In: IEEE international conference on web services, pp 856–863
10. The Eclipse Foundation (2010) Eclipse IDE. http://www.eclipse.org
11. The Eclipse Foundation (2010) Model development tools (MDT). http://www.eclipse.org/modeling/mdt/?project=xsd
12. W3C (1999) HTML 4.01 specification. http://www.w3.org/TR/html401
13. W3C (2000) Document object model (DOM) level 2 core specification. http://www.w3.org/TR/DOM-Level-2-Core

14. W3C (2002) XHTML 1.0: The extensible hypertext markup language, 2nd edn. http://www.w3.org/TR/xhtml1
15. W3C (2004) XML schema. http://www.w3.org/XML/Schema
16. W3C (2009) The extensible stylesheet language family (XSL). http://www.w3.org/Style/XSL
17. W3C (2009) XForms 1.1. http://www.w3.org/TR/xforms11
18. World Wide Web Consortium (1999) XSL transformations (XSLT), version 1.0. http://www.w3.org/TR/xslt

Research Directions of OLAP Personalizaton

Natalija Kozmina and Laila Niedrite

Abstract In this paper we have highlighted five existing approaches for introducing personalization in OLAP: preference constructors, dynamic personalization, visual OLAP, recommendations with user session analysis and recommendations with user profile analysis and have analyzed research papers within these directions. We have provided an evaluation in order to point out i) personalization options, described in these approaches, and its applicability to OLAP schema elements, aggregate functions, OLAP operations, ii) the type of constraints (hard, soft or other), used in each approach, iii) the methods for obtaining user preferences and collecting user information. The goal of our paper is to systematize the ideas proposed already in the field of OLAP personalization to find out further possibility for extending or developing new features of OLAP personalization.

1 Introduction and Related Work

The OLAP applications are built to perform analytical tasks within large amount of multidimensional data. During working sessions with OLAP applications the working patterns can be various. Due to the large volumes of data the typical OLAP queries performed via OLAP operations by users may return too much information that sometimes makes further data exploration burdening or even impossible. In case of too many constraints chosen the result set can be empty. In other cases, when the user tries to explore previously unknown data, the OLAP query result may highly differ from expectations of the user. Thus, the user is rather limited in expressing his/her intentions or likes and dislikes in order to get more satisfying results.

A query personalization method that takes user likes and dislikes into consideration exists in traditional databases [1]. So, in case of executing a personalized query, the user gets more appropriate results. Similar ideas seem attractive also for research in the data warehousing field and the topicality of this issue is demonstrated in the recent works of many authors on data warehouse personalization.

There are various aspects of data warehouse personalization.

N. Kozmina (✉)
Faculty of Computing, University of Latvia, Riga LV-1586, Riga, Latvia
e-mail: natalija.kozmina@lu.lv

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_28,

Data warehouse can be personalized at the schema level. [2] use the data warehouse multidimensional model, user model and rules for the data warehouse personalization. As a result, a data warehouse user is able to work with a personalized OLAP schema, which matches his needs best of all.

Users may express their preferences on OLAP queries [3]. In this case, the problem of performing time-consuming OLAP operations to find the necessary data can be significantly improved.

One of the methods of personalizing OLAP systems is to provide query recommendations to data warehouse users. OLAP recommendation techniques are proposed in [4, 5]. In [4] former sessions of the same data warehouse user are being investigated. User profiles that contain user preferences are taken into consideration in [5], while generating query recommendations.

Other aspect of OLAP personalization is visual representation of data. In [6, 7] authors introduce multiple layouts and visualization techniques that might be interactively used for different analysis tasks.

Our experience in using standard applications for producing and managing data warehouse reports in the University of Latvia (UL) as well as participation in scientific projects and development of our own data warehouse report management tool [8] served as a motivation for further studies in the field of OLAP personalization. It has been stated that both tools (standard and newly-developed) allow defining ad-hoc queries, displaying reports as tables and graphs and analyzing data using hierarchies. Users with administrator rights may modify other user right for data warehouse report creating, exploring and editing. A user may adjust visual representation of the workbook, which contains generated reports (e.g. change font color and style, etc.). Since options to personalize data warehouse reports by means of these tools are currently very limited, we consider the report management tool, developed in the UL, to be an experimental environment for introducing OLAP personalization.

As stated in [3], OLAP preferences deserve more attention by researchers. In this paper an overview of different OLAP personalization approaches is presented. The goal of our paper is to classify the ideas that have been already proposed in this field in order to find questions that still remain unanswered.

The rest of the paper is organized as follows: Sect. 2 introduces a review of existing OLAP personalization types and its evaluation; Sect. 3 discusses hard and soft constraints in user preferences as well as methods for gathering user information and obtaining user preferences; Sect. 4 concludes the paper.

2 OLAP Personalization Types

To the best of our knowledge, there are various OLAP personalization types—OLAP schema personalization, personalization during runtime, visual personalization of query results, etc.—which are briefly described in this section. A comparison, which includes personalization types and OLAP schema elements and operations, will follow. Proposed comparison gives our evaluation of personalization described by indicating, whether personalization of certain type is applicable to OLAP schema elements and operations, or not.

2.1 Description of OLAP Personalization Approaches

The first approach to be considered is OLAP schema personalization with *Preference Constructors (PC)*. An algebra that allows formulating of preferences on attributes, measures and hierarchies is defined in [3]. An important feature of proposed algebra is an opportunity to express preferences for hierarchy attributes of group-by sets, which consequently leads to expressing preferences for facts. Rollup function is used to outspread preferences applied to attributes along the whole hierarchy. Preferences can be defined on both attributes and measures, i.e. on categorical or numerical attributes.

Consider two kinds of preferences: base and complex [3]. Base preference constructors are applied to attribute, measure, hierarchy level. Complex preferences consist of combination of base preferences, which can be expressed by means of formal grammar. Base preference constructor in this grammar is one of predefined operators like POS, NEG, BETWEEN or some others.

The next approach is *Dynamic Personalization (DP)*. The time and method of creation of an adapted OLAP cube define the type of personalization—static or dynamic. Static OLAP personalization means that for different users of the data warehouse diverse OLAP cubes are created during design time. Dynamic OLAP personalization means that an adapted OLAP cube is created during the execution time according to the needs and performed actions of the user. Authors [2] cover dynamic OLAP personalization, because it is a more complicated task as it involves explicit or implicit interaction with user. Based on ECA-rules (see [9]), PRML (described in [10]) is used in [2] for specification of OLAP personalization rules. The structure of such PRML rule can be presented with following statement:

```
when event do if condition then action endIf endWhen.
```

There are two kinds of actions proposed to be used in personalization rules in [2]. In order to get information about the user during runtime and update the user model or to update values of dimension attributes and cube measures, a *set*-action is used (e.g. for calculating user's degree of interest in certain dimension attributes). To personalize multidimensional model, *hide*-actions are used on OLAP schema objects (e.g. a *hide*-action may be executed, if the user's degree of interest in a certain dimension attribute is lower than a pre-defined value).

Visual personalization of OLAP cube—*Visual OLAP (VO)*—may also be considered as a personalization action. The concept of Visual OLAP is disburdening the user from composing queries in "raw" database syntax (SQL, MDX), whereas events like clicking and dragging are transformed into valid queries and executed [7]. In [5, 6, 11] authors present a user interface for OLAP, where user is explicitly involved. In [6] users are able to navigate in dimensional hierarchies using a schema-based data browser, whereas in [5, 11] users are provided with an interface for formulating queries by means of manipulation with graphical OLAP schema and rules. The query is composed by the user when he/she selects a measure and an aggregation function [6]. Dimensions for "drilling down" are chosen and the values are set as filters. Having selected the measure and the aggregate function, the user simply drags any dimension folder into the visualization area to create a new level in the *decomposition tree*. The decomposition tree is gained from an aggregate measure as a root, splitting it along chosen dimensions. Different layouts for decomposition trees are proposed in [6].

The last two approaches for personalization in OLAP to be considered are based on providing query recommendations to the user by means of *User Session Analysis (RUSA)* and *User Preference Analysis (RUPA)*.

The idea of *RUSA* is described in [4], where users' previous data analysis patterns using OLAP server query log during sessions are taken into consideration. Cube measure values are being compared and a significant unexpected difference in the data is being detected. The emphasis is not on recommending queries from sessions that are prior to the current session, but on recommending queries from all sessions, where user found the same unexpected data as in current session. In this approach user preferences are not taken into consideration. A concept of difference query for rollup and drill-down operations as a query whose result confirms the difference of measure values at a higher level of detail for rollup or lower level of detail is introduced by [4]. Authors analyze user queries, executed during users' sessions, thus we consider that personalization is applicable to OLAP select operation.

RUPA approach is presented in [5], where a context-based method for providing users with recommendations for further exploration is proposed. An analysis context includes two disjoint set elements (i.e. a set of OLAP schema elements—cubes, measures, dimensions, attributes, etc. and a set of its values), which are represented in a tree structure (though visualized as a multidimensional table). Also, restriction predicates i.e. restrictions on measures (associated with an aggregate function) or conditions on dimension attributes are included into analysis context. Both types of user preferences—schema- and content-level preferences—are stated in the user profile and ranked with relevance score (a real number in the range [0; 1]). The idea of ranking preferences is also mentioned in [11]. User preferences later on are used in generating recommendations, filtering a recommendation with the highest overall score and displaying it to the user. Preferences in user profiles are also used for comparing queries and personalizing query result visualization in [12].

2.2 Comparison of Existing Approaches for OLAP Personalization

We analyzed and compared all previously described approaches to give an overview on applying personalization of different type to OLAP schema elements, functions and typical OLAP operations. The results are given in Table 1. One axis of the table contains the main concepts of OLAP systems: OLAP schema elements, aggregate functions, OLAP operations. The OLAP schema elements—dimensions and its attributes, hierarchies and its levels, cubes (or fact tables) and its measures—are described a lot in the literature, also in [13]. Aggregate functions are described in [14]. OLAP operations slice and dice (or select), drilldown, rollup and pivot (or

Table 1 Applicability of different personalization types to OLAP objects

Pers. Type / Pers. Object	Dimension	Dimension attribute	Hierarchy	Hierarchy level	Cube	Cube measure	Aggregation function	Select	Drilldown	Rollup	Rotate
PC	-	*A*	-	*A*	-	*A*	-	*D*	*D*	*A*	-
DP	*A*	*A*	*A*	*A*	*A*	*A*	*A*	*A**, *D***	*A*	*A*	*A**, -**
VO	*A*	*A*	*A*	*A*	*A*	*A*	*A*	*A*	*A*	*A*	*A*
RUSA	*A*	*A*	*A*	*A*	*A*	*A*	*D*	*A*	*A*	*A*	-
RUPA	*A*	*A*	*A*	*A*	*A*	*A*	*A*	*A*	*D*	*D*	-

rotate) are described in [13, 15, 16]. In our comparison we use a term *select* instead of *slice and dice* for the sake of simplicity, because some of the personalization types provide personalization of SQL-like select-queries. Also, here we use the term *rotate* instead of *pivot*. The second axis of the table contains all previously described personalization types. The cells of the table contain a value from a set of acronyms to represent our evaluation: "A"—applicable: personalization applicability to OLAP schema element, aggregate function or OLAP operation is explicitly defined by the authors of articles on *PC*,*DP*, *VO*, *RUSA* and *RUPA*; "D"—derivable: personalization applicability to OLAP schema element, aggregate function or OLAP operation can be derived, taking into account other personalization aspects, which are presented in the paper (e.g. personalization considers rollup operation, but drilldown operation is not mentioned in the paper; in that case we say that personalization considering drilldown is *derivable*, because drilldown operation is an inverse operation of rollup, etc.); "–"—there is no information; personalization applicability to OLAP schema element, aggregate function or OLAP operation is not described in the paper.

In *DP* the extent of personalization applied to certain OLAP operations varies, depending on the approach, proposed by different authors. Authors [11] are marked with "*", authors [2]—with "**".

One may observe that personalization of OLAP schema elements is mostly present in all proposed OLAP personalization types, except for preference constructors (*PC*), where the way of expressing user preferences for dimensions, hierarchies, cubes as whole as well as aggregate functions, is not described. However, preferences on OLAP operations such as Select, Drilldown and Rollup are not always expressed explicitly and there is a lack of information about personalization options, considering Rotate OLAP operation.

3 A Closer Look at User Preferences: Hard and Soft Constraints

Although the role of the preferences was recognized in applications long ago, the database researchers paid attention to this issue only around year 2000 [17–20]. It was observed that in database queries WHERE-conditions are *hard constraints* and either the non-empty result set is returned if all the conditions are satisfied, or an empty set is returned in the opposite case. Queries with hard constraints either deliver exactly the desired object if it exists, or reject the user's request otherwise [21].

The authors of [22] define *soft constraints* as functions that map any potential value assignment into a numerical value that indicates the preference that this value or value combination carries. In information retrieval soft constraints are used and results are arranged, according to its relevancy to initial query conditions. The difference between hard and soft constraints is that soft constraints can be evaluated, whereas hard constraints can be either satisfied or not. User preferences express soft constraints. Eventually, different approaches to use soft constraints in database queries have appeared [19, 20], turning database queries into "preference queries". In papers [19, 21, 23, 24] an implementation of the framework using Preference SQL is described, which is translated to SQL, and used in several deployed applications. Kießling [19] and Chomicki [21] point out that extending SQL by preferences will enable better personalized search to gain more targeted results.

Preference SQL consists of Standard SQL constructs and preferences [24]. Preference queries are specified in [23] using a SELECT-FROM-WHERE part (standard SQL; the WHERE-clause specifies hard constraints) and a PREFERRING-GROUPING part (expresses preferences i.e. soft constraints) of a query. In both parts of a preference query AND can be used to combine more than one constraint, but in the PREFERRING clause it has a meaning of Pareto operator. In this case AND prescribes combination of equally important preferences.

Our purpose is to understand, what kind of user preferences can be expressed in each of OLAP personalization types earlier discussed. We consider the hard and soft constraints as a means to express the user preferences. In the OLAP domain the definition of *preference* is proposed by [3], stating that a preference is a couple of two operators, where first states that one fact value in OLAP schema is preferred to the other, but the other operator states that facts are equivalent (or *substitutable* [25]).

Table 2 illustrates, which method is applied; " +" ("−") indicates that a method is (isn't) applied in each of personalization types: Hard/Soft Constraints or Other (meaning that the method used cannot be categorized as hard/soft constraints).

Preference Constructors (*PC*) use soft constraints as there is a possibility to express user's likes and dislikes, e.g. a user would like to obtain student activity data (i.e. time spent on exploring course informational resources, quantity of tasks assigned and completed, grades for completed tasks, etc.) considering course, named "*Data Warehouses*", which is an attribute of Course dimension in data warehouse.

Table 2 OLAP Personalization types and applied constraints

Personalization Type / Method	Hard Constraints	Soft Constraints	Other
PC	-	+	-
DP	+	-	-
VO	+	-	-
RUSA	-	-	+
RUPA	-	+	-

Example 1. Consider a hierarchy Course $\succ_H$ Study Program $\succ_H$ Faculty, where $\succ_H$ is a Rollup function over hierarchy H. *Biology Masters* is one of study programs, belonging to the *Faculty of Biology*. NEG(StudyProgram, "Biology Masters") states that data that does not map to *Biology Masters* study program, does not refer to courses of *Biology Masters* study program and does not map to the *Faculty of Biology*, is preferred to all the other data.

One of the aspects of Visual OLAP (*VO*) is user browsing through navigational OLAP schema and filtering the OLAP schema objects to be displayed [7]. Users' navigation events such as clicking and dragging are translated to valid SQL-queries with WHERE-clause, which in fact is a hard constraint in standard SQL [21].

We consider that there are hard constraints in dynamic personalization (*DP*) with ECA-rules as the sets of operations with both numerical and non-numerical attributes in condition-part of ECA-rules are the same as operations, included in hard constraints. In the following example " =" operation is used when checking, whether the data warehouse user role is "Student" or not; if the user is a student, then attribute *BusinessTrip* of the dimension *Person* is being hidden.

Example 2. **Rule**: hideBusinessTrip
When SessionStart **Do If** (User.Role = "Student")
Then hideDescriptor(Person.BusinessTrip) **EndIf EndWhen**

The main idea of query recommendations approach, based on investigation of user sessions (*RUSA*), is to find unexpected difference in the data and generate further recommendations with the same unexpected data as the current session.

Example 3. If there is a difference that is a drop of the sales of some kind of product from 2009 to 2010, then recommended queries will contain the same difference in values. We consider that neither soft nor hard constraints are used in this type of personalization. Authors [4] use the technique that develops the ideas of DIFF operator, proposed in [26] and used for explaining reasons for sudden drops or increases in data values.

We consider that in user profiles, utilized for generation of recommendations (*RUPA*), soft constraints appear. A user may express the extent of liking or disliking as there is a relevance score that is associated with analysis element of OLAP schema [5]. Following example illustrates the usage of soft constraints in *RUPA*:

Example 4. P^{Role} = ('Role $\neq$ Guest'; 0.9; c), where 'Role $\neq$ Guest' is a predicate, which is a condition on dimension data (in other case a predicate may be a restriction on fact table data), 0.9 is a real number (between 0 and 1) that indicates relevance degree (a number closer to 0 means 'less relevant' and closer to 1 means 'more relevant'), c is an analysis context, which includes analyzed cube measures (with aggregate functions applied) and analysis axis (dimension/attribute). Here c = "Activity, Time/Date $\geq$ '01/01/2010"', which means that measures of Activity cube are analyzed and Time/Date is an analysis axis. P^{Role} = ('Role $\neq$ Guest'; 0.9; c) means that user's interest to include condition 'Role $\neq$ Guest' into qualification of user activity in course management system is very high.

3.1 Collecting User Data: Explicit and Implicit Approaches

Typically there are two approaches of collecting information about the user—explicit and implicit feedback [27]. Also, hybrid (i.e. explicit and implicit method combined) is possible.

Methodologies for *explicit* user information gathering are based on information input by users about themselves and their interests. Users enter information manually or choose pre-defined values from the list. Problems arise, because users are not always ready to give such information. In this case an explicit user profile could not be built. Also [27], points out that a user may not be very accurate, when providing information. User preferences may change over time, thus information in the profile may become out of date.

User profiles may be built based on *implicitly* gathered information. Implicit feedback gives us behavioral information about the user. Implicit feedback can be found by analyzing server logs, search and browsing history. A research on acquiring user preferences, based on implicit feedback, is presented in [28].

The most attractive aspect of the implicit feedback is that data about the user can be gathered without the presence of the user [27]. However, authors [27] point out some limitations of the implicit feedback. The data, observed by the user, is not always connected with an intention to observe it. Often the time when the data is displayed to the user is interpreted as reading time. Also, the user is unable to give negative feedback, to express negative interest or dislike, whereas mouse clicks are treated as positive feedback [27]. Sometimes during the search for essential information user clicks on unnecessary links, therefore, in many cases user activity could not be equalized to the count of clicks.

3.2 Methods for Obtaining User Preferences

Author [29] gives an overview of existing methods for extracting user preferences and giving further recommendations; authors [30] supplement the list with two more methods (*questions & answers*, *mixed initiative*):

1. *Questions & Answers (Q&A)*. Information for user profile is collected, when user answers to the questions or fills in the form. The information in user profile stays unchanged, until the user updates it.
2. *Mixed initiative (MI)*. This method is also called *candidate/critique mode*. Preferences are gained by proposing existing solutions to a user and receiving user evaluation. The solution is improved, according to the critique and proposed to the user again until it satisfies the user. An example of a system with implemented mixed initiative approach is a system, presented by [31], where an agent is implemented for the gathering user preferences when the user expresses his/her attitude to the observed data.
3. *Content-based (CB)*. This method is used to generate recommendations from user preferences on other objects' features that user has already rated. Content-based user profiles are updated, when new user preference-related information appears.

Table 3 Preference obtaining and user information collection methods, used in different types of OLAP personalization

	Preference Obtaining Method						*User Information Collection Method*	
Personalization Type	Q&A	MI	CB	UKB	C	D	Ex-plicit	Implicit
PC	+	-	-	+	-	-	+	-
DP	-	-	+	+	-	-	-	+
VO	-	-	+	-	-	-	+	-
RUSA	-	-	-	+	-	-	-	+
RUPA	-	-	+	-	-	-	+	-

4. *Utility and Knowledge-based (UKB)*. These methods make recommendations, based on similarity between what user needs and what is available.
5. *Collaborative (C)*. In terms of this method multiple user ratings are aggregated and compared with the rating of a particular user of a certain object. As a result new recommendations are proposed to the user.
6. *Demographic (D)*. This method is used to provide recommendations based on demographic characteristics of the user. Users with similar characteristics are grouped into classes.

Table 3 illustrates, which preference obtaining method is applied in each of five considered OLAP personalization approaches as well as demonstrates, how user information was collected—explicitly or implicitly.

Although in [3] OLAP preference algebra is proposed and technical implementation of preference constructors (*PC*) and its application is not described, we consider that the user would express the preferences explicitly. For instance, user may choose some out of the set of possible preference constructors and OLAP schema elements that serve as parameters for preference constructors, and assign values for OLAP schema elements (entering manually or choosing from range). Such approach is similar to *Q&A* method. Also, *UKB* method is being partly used, when, for instance, user states a certain attribute value in POS or NEG constructor, and then preferences are propagated over all levels of the corresponding hierarchy (see Example 1).

We suppose that there is a content-based (*CB*) approach used in dynamic personalization (*DP*). For instance, when ECA-rules are being executed, user context is taken into consideration e.g. user role in data warehouse (see Example 2). Also, *UKB* approach is used, when user behavior is being analyzed, for instance, a utility is used for calculating user interest degree in certain aggregated data. Dynamic personalization uses an implicit method for collecting user information.

A content-based approach is also used in visual OLAP (*VO*) and in recommendations with user profile analysis (*RUPA*). In *VO* the user is able to move through navigational schema and set preferences for OLAP schema objects to be displayed (for example, choosing dimensions, setting constraints on dimension attribute values, etc.). In *RUPA* schema- and content-level preferences are stated in a user profile and ranked with relevance score. In both cases information is provided explicitly by the user.

In recommendations with user session analysis (*RUSA*) user information is gathered implicitly. To define user preferences, *UKB* approach is used—user previous session queries are being examined and a utility function, conceptually similar to DIFF operator, is applied (see Example 3).

4 Conclusions and Future Work

In this paper we have highlighted five approaches for introducing personalization in OLAP: preference constructors (*PC*), dynamic personalization (*DP*), visual OLAP (*VO*), recommendations with user session analysis (*RUSA*) and recommendations with user profile analysis (*RUPA*). We do not claim that this is an exhaustive set of approaches for OLAP personalization and assume that it may be widened.

We have provided an evaluation in order to point out (i) personalization options, described in these approaches, and its applicability to OLAP schema elements, aggregate functions, OLAP operations, (ii) the type of constraints (hard, soft or other), used in each approach, (iii) the methods for obtaining user preferences and collecting user information.

Comparing options of personalization application to personalization types, we may conclude that personalization of OLAP schema elements is mostly present in all proposed OLAP personalization types, except for preference constructors (*PC*), where the way of expressing user preferences for dimensions, hierarchies, cubes as whole as well as aggregate functions, is not described. Speaking about OLAP operations, we may notice that three out of four OLAP operations in three out of five personalization types are described implicitly (i.e. Select and Drilldown operations in *PC*, Select operation in *DP*, Drilldown and Rollup operations in *RUPA*). The information about expressing user preferences on Rotate operation is missing in all approaches, except for *DP* and *VO*. Thus, more attention should be drawn to user preferences for OLAP operations.

We proposed to group personalization types, according to the kind of constraint (soft, hard or other) that is used for expressing and managing user preferences. As a result, hard constraints are used in *DP* and *VO*, soft constraints—in *PC* and *RUPA* and other type of constraint (difference function)—in *RUSA*.

We analyzed applicability of existing methods for extracting user preferences [29, 30] and highlighted how the user information is being collected (explicitly or implicitly). We may conclude that three out of six preference obtaining methods (i.e. questions & answers, content-based and utility & knowledge-based) are applied in considered types of personalization and the remaining three methods (mixed initiative, collaborative and demographic) are not applied. However, we assume that it is worthwhile to involve collaborative method for generating recommendations of queries, based on similarity of users' likes and dislikes.

We have taken the ideas of *RUPA* approach as a basis for our future work. We also proposed to involve collaborative method for generating recommendations of queries, based on similarity of users' likes and dislikes. A new method, which provides exhaustive description of interaction between user and data warehouse, is a subject of a separate paper.

Acknowledgements This work has been supported by ESF project No.2009/0216/1DP/1.1.1.2.0/09/APIA/VIAA/044.

References

1. Koutrika G, Ioannidis YE (2004) Personalization of queries in database systems. In: Proceedings of 20th international conference on data engineering (ICDE'04), Boston, 30 Mar–2 Apr 2004, pp 597–608
2. Garrigós I, Pardillo J, Mazón J-N, Trujillo J (2009) A conceptual modeling approach for OLAP personalization. In: Conceptual modeling—ER 2009, LNCS 5829. Springer, Heidelberg, 401–414
3. Golfarelli M, Rizzi S (2009) Expressing OLAP preferences, LNCS 5566/2009. Scientific and Statistical Database Management, Berlin/Heidelberg, pp 83–91
4. Giacometti A, Marcel P, Negre E, Soulet A (2009) Query recommendations for OLAP discovery driven analysis. In: Proceedings of 12th ACM international workshop on data warehousing and OLAP (DOLAP'09), Hong Kong, 6 Nov 2009, pp 81–88
5. Jerbi H, Ravat F, Teste O, Zurfluh G (2009) Preference-based recommendations for OLAP analysis. In: Proceedings of the 11th international conference on data warehousing and knowledge discovery (DaWaK'09), Linz, Austria, 31 Aug–Sept 2009, pp 467–478
6. Mansmann S, Scholl MH (2007) Exploring OLAP aggregates with hierarchical visualization techniques. In: Proceedings of 22nd annual ACM symposium on applied computing (SAC'07), Multimedia and visualization track, Mar 2007, Seoul, Korea, pp 1067–1073
7. Mansmann S, Scholl MH (2008) Visual OLAP: a new paradigm for exploring multidimensonal aggregates. In: Proceedings of IADIS international conference on computer graphics and visualization (MCCSIS'08), Amsterdam, The Netherlands, 24–26 July 2008, pp 59–66
8. Solodovnikova D (2007) Data warehouse evolution framework. In: Proceedings of the spring young researcher's colloquium on database and information systems SYRCoDIS, Moscow, Russia. http://ceur-ws.org/Vol-256/submission_4.pdf
9. Thalhammer T, Schrefl M, Mohania M (2001) Active data warehouses: complementing OLAP with active rules. Data Knowl Eng 39(3):241–269, Dec 2001, Elsevier Science Publishers B. V., Amsterdam, The Netherlands
10. Garrigós I, Gómez J (2006) Modeling user behaviour aware websites with PRML. In: Proceedings of the CAISE'06 3rd international workshop on web information systems modeling (WISM'06), Luxemburg, 5–9 June 2006, pp 1087–1101
11. Ravat F, Teste O (2009) Personalization and OLAP databases. Ann Inf Syst 3:1–22, New Trends in Data Warehousing and Data Analysis, Springer US
12. Bellatreche L, Giacometti A, Marcel P, Mouloudi H (2006) Personalization of MDX queries. In: Proceedings of XXIIemes journees Bases de Donnees Avancees (BDA'06), Lille, France
13. Kimball R, Ross M (2002) The data warehouse toolkit: the complete guide to dimensional modeling, 2nd edn. Wiley, New York
14. Lenz H-J, Thalheim B (2009) A formal framework of aggregation for the OLAP-OLTP model. J Universal Comput Sci 15(1):273–303
15. Inmon WH (2002) Building the data warehouse, 3rd edn. Wiley Computer Publishing, New York, 428 p
16. Adamson C (2006) Mastering data warehouse aggregates: solutions for star schema performance. Wiley Computer Publishing, New York, 384 p
17. Agrawal R, Wimmers E (2000) A framework for expressing and combining preferences. In: Proceedings of the ACM SIGMOD international conference on management of data. ACM, New York, pp 297–306
18. Borzsonyi S, Kossmann D, Stocker K (2001) The skyline operator. In: Proceedings of 17th international conference on data engineering, Heidelberg, April 2001

19. Kießling W (2002) Foundations of preferences in database systems. In: Proceedings the international conference on very large databases (VLDB'02), Hong Kong, China, pp 311–322
20. Chomicki J (2003) Preference formulas in relational queries. ACM TODS 28(4):427–466
21. Kießling W, Köstler G (2002) Preference SQL-design, implementation, experiences. In: Proceedings of the international conference on very large databases (VLDB'02), Hong Kong, China, pp 990–1001
22. Pu P, Faltings B, Torrens M (2003) User-involved preference elicitation. In: IJCAI'03 workshop on configuration, Acapulco, Mexico
23. Hafenrichter B, Kießling W (2005) Optimization of relational preference queries. In: Proceedings of the 16th Australasian database conference, ADC 2005, vol 39, Newcastle, Australia, 31 Jan–3 Feb 2005, pp 175–184
24. Kießling W (2006) Preference handling in database systems. Talk at L3S, University of Hannover, 6 Feb 2006
25. Kießling W (2005) Preference queries with SV-semantics. In: Proceedings of COMAD'05, Goa, India, pp 15–26
26. Sarawagi S (1999) Explaining differences in multidimensional aggregates. In: Proceedings of the international conference on very large databases (VLDB'99), 7–10 Sept 1999, Edinburgh, Scotland, UK, pp 42–53
27. Gauch S, Speretta M, Chandramouli A., Micarelli A (2007) User profiles for personalized information access. In: Brusilovsky P, Kobsa A, Nejdl W (eds) The adaptive web (Chap. 2), LNCS 4321. Springer, Berlin, pp 54–87
28. Kelly D, Teevan J (2003) Implicit feedback for inferring user preference: a bibliography. ACM SIGIR Forum 37(2):18–28
29. Burke R (2002) Hybrid recommender systems: survey and experiments. User Model User Adapted Interact 12(4):331–370, Kluwer Academic Publishers, Norwell
30. Viappiani P, Pu P, Faltings B (2002) Acquiring user preferences for personal agents. Technical report for American Association for Artificial Intelligence (AAAI Press). http://liawww.epfl.ch/Publications/Archive/Viappiani2002.pdf
31. Shearin S, Lieberman H (2001) Intelligent profiling by example. In: Proceedings of IUI'01, Santa Fe, New Mexico, USA, 14–17 Jan 2001, pp 145–151

Extending UML with Non-functional Requirements Modelling

Aneesh Krishna and Andreas Gregoriades

Abstract Non-Functional Requirements (NFR) have been for too long overlooked during the development of software systems, leading in numerous cases to failure, an over budget or even cancellation of projects. In this paper, we propose a solution to integrate the modelling of NFRs into the UML diagrams. Our framework is inspired by several proposals in this critical area, but it is the first time that a unified process, allowing possible integration of NFRs in necessary UML diagrams, is proposed.

1 Introduction

For a software development project to be successful, the key is to elicit and understand the requirements from the beginning of the project. Functional Requirements (FRs) determine what functions a software product should offer. Multiple proposals have been developed to analyze and model them, and are nowadays widely used. For Non-Functional Requirements (NFRs), the problem is more complex and due to that they have been ignored for a long time in the early phases of the software development process. NFRs are defined as constrained on the emergent properties of the overall system. Hence, software products are sometimes refined late in the development process to satisfy NFRs, this as a result leads to over budget or cancelled projects. One such example is the London Ambulance System (Finkelstein and Dowell 1996). Software engineers now acknowledge the importance of dealing with NFRs in the early phases of the development process (Russell 2004), however, there is lack of credible proposals to be used in the software development. In this paper we propose a solution to integrate the analysis and modelling of NFRs into UML (OMG 2007a, b), which is the most popular language for modelling purposes in object-oriented paradigm. Our goal is to offer a method to analyze, refine and integrate NFRs in existing diagrams to provide justifications for the design.

A. Krishna (✉)
Department of Computing, Curtin University of Technology, Perth, Australia
e-mail: a.krishna@curtin.edu.au

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_29,

This paper is structured as follow: Sect. 2 describes the background and related work in the area. Section 3 formally formulates the proposal, providing new notations to deal with NFRs through an example. Finally, Sect. 4 concludes the paper.

2 Background and Related Work

Chung et al. (1995, 2000) developed a framework to represent and analyze non-functional requirements. This framework is an interesting one; having potential to fully analyze the dependencies between non-functional requirements. This work uses the concept of 'claim' in the diagram. Claims are linked to interdependency, and provide a rationale to design decision. However, having multiple claims can make diagrams complicated and harder to read. Plus, design rationales are easy to document outside the diagram. Cysneiros et al. (2001) first proposed a method to integrate NFRs into ER and UML data models. The process was refined by Cysneiros and do Prado Leite (2001). This strategy uses the notion of lexicon, namely, the Language Extended Lexicon (LEL), as an anchor to integrate NFRs into OO models. The idea of LEL is to register the vocabulary (symbols) of a project, and determine the relations between the symbols. In their proposal, Cysneiros et al. (2001) extended the LEL to deal with NFRs. The solution proposed herein is inspired from that method. The concept of LEL is not widely accepted/used in software industry. Imposing its usage would hugely increase the workload of the systems analysts. Therefore, another solution is needed to link NFRs to UML diagrams that would have practical usefulness in industry. This work also considers that functional and non-functional aspects should be carried through two independent cycles with convergence points, while efforts should be carried out to merge the two aspects. Cysneiros and do Prado Leite (2001)'s proposal can be used to integrate NFRs into any UML diagram, but no support is provided to deal with diagrams other than Class Diagrams. These reasons form the basis for our proposal in this paper.

3 Proposal for Integrating Non-functional Requirements into UML Diagrams

3.1 Overview of the Proposed Strategy

The proposal considers that NFRs and FRs should not be carried out through two independent cycles, but should be part of a common, global strategy. This decision had a huge impact on the process designed. The process can be visualized in Fig. 1.

The main steps of the proposal are the following:

1. First, we determine what the FR and NFRs are.
2. From the FRs, we shall draw the use cases, and then associate the NFRs with the different parts of the system they control. When associating them, two cases may

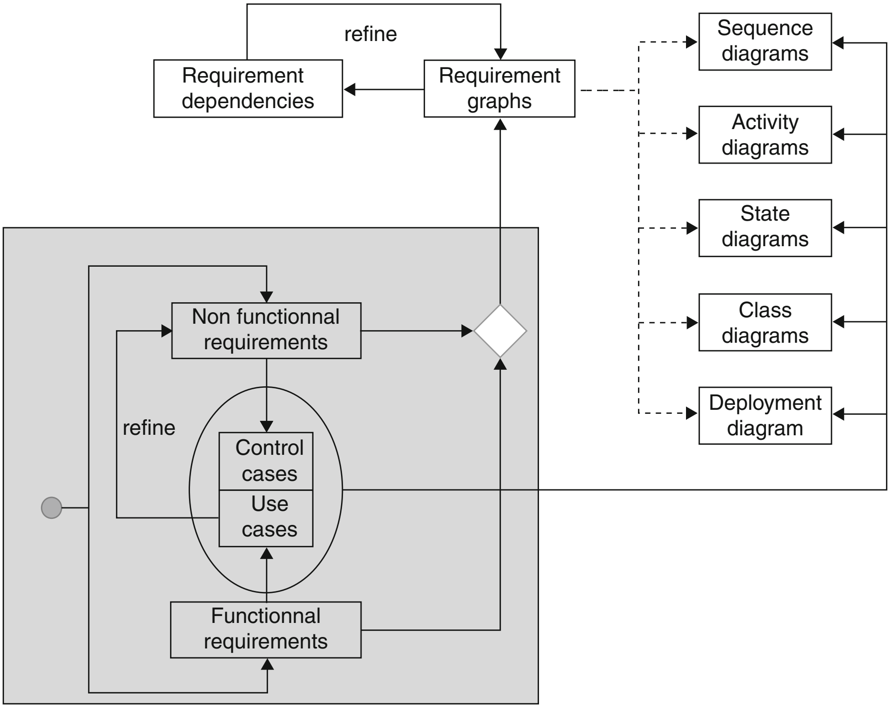


Fig. 1 Diagram of the proposed strategy

arise: either the NFR is specific to a FR (e.g. 'the authentication must be fast'), or it is global and thus not linked to a single FR (e.g. 'all screens of the system should look similar'). In the first case, the NFR is linked with a use case, else with the entire system.

3. The NFRs will then be refined using the use cases. In this step, the systems analysts will use his experience to find missing NFRs that should be included in the diagrams. This is quite important, because the stakeholders do not always think about every single NFR (some might seem obvious, other irrelevant at first), or some of them are simply hard to come up with. This should prevent NFR from emerging at later stages of the project.
4. Using the FRs and NFRs that have been extracted, a requirement graph will be drawn. This graph will show the decompositions of the NFRs into subgoals, how to satisfy the sub-goals with operationalizations, and the way all these elements interact with each other. The focus here is solidly on FR that can impact NFRs, so not every FR will be included in this graph.
5. Study the interdependencies between the different requirements, their subgoals and their operationalizations. If necessary, do trade offs (for example, using uncompressed format is good for performance, but not for space usage), and more generally, refine the requirements graph by studying the impact of each interdependency on the related requirements (see Sect. 3.4).

6. Finally, use all the information gathered to draw other UML diagrams. This paper highlights only the application of this to sequence, class and deployment diagrams.

3.2 Use Cases Controlled by NFRs

Incorporation of NFRs into use case diagrams is done by using a solution similar to Zou and Pavlovski (2006). A prerequisite for this is a complete or near complete list of the system requirements. FR are subsequently used to produce use cases. Each NFR is documented with three pieces of information: the interested stakeholder, the NFR's type and NFR identification code (ID). Representing a NFR in the diagram these three information is particularly important. First, we need a non ambiguous way to link a use case to a NFR, which is done by stating the ID of the NFR. Then, including both the stakeholder and the type of the NFR in the representation provides the systems analysts with useful information without overloading the schema. In fact, the system analysts will not need to read the exact object of the NFR to know how the control will impact on the use case: mentioning the type of the NFR related is usually enough (e.g. a NFR of type 'security' on an authentication operation is self-explanatory). Contrary to Zou and Pavlovski (2006) the approach does not include the 'control target' of the NFRs on this diagram, this information is not useful here as we are not concerned about the implementation at this point in time.

Like stated earlier, two types of control can appear: specific and global. Specific control applies to a particular use case, and as such its represented inside the system. Global control applies to the whole system, and is represented outside the system. The general representation of such a diagram is presented on Fig. 2.

Once the diagram has been drawn, it is fairly easy for the system analysts to go through the different use cases and see, for each one, the types of the NFRs related. This will help to find missing NFRs, either through his personal experience or by using check lists. For example, an authentication use case with no security NFR related is obviously wrong.

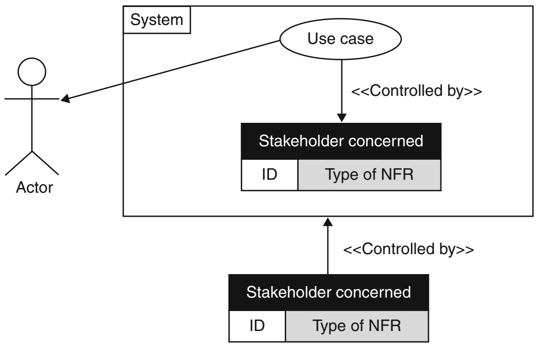


Fig. 2 Actors and use cases controlled by NFRs

3.3 Construction of the NFRs Graph

The aim of this step is to decompose a NFR into sub-softgoals (Chung et al. 2000). Then, for sub-softgoals that can't be subdivided find alternatives to operationalize it. Therefore, the following process has to be completed for each NFR represented in the use cases/control cases diagram. Add the object that the NFR is related to. Given that the NFR is not linked to a use case in this diagram, it is important to know on what the NFR acts (e.g. if we consider the informal high-level NFR 'Good Performance for accounts', the object is 'Account'). Next, the analyst has to specify the priority of the NFR. This is important to algorithmically determine what subset of the NFRs can actually be satisfied. The mark 1 means that the NFR can be easily ignored, while 5 mean that the NFR has to be satisfied no matter what. Determining the priority of a NFR is not always easy as each stakeholder has his own views on the project, but can be helped by doing a stakeholders' analysis (Smith 2000). Subsequently, the analyst adds the control target that designates if the NFR is somehow quantifiable. This helps to decompose the NFR into sub-softgoals and later find operationalizations. With the conclusion of the process the NFR is renamed to 'Formulated NFR' as shown in Fig. 3.

Once the formulated NFR has been added on the diagram, the systems analysts has to divide it into sub-softgoals (see Fig. 4) linked to the formulated NFR by the connectors shown on Fig. 5. For each sub-softgoal, the systems analysts has to indicate the refinement compared to the parent NFR (but one should note than the Type of NFR can change as well, e.g Perfomance can be divided to Space and Time). In our proposal, the priority of a sub-softgoal is always inherited from the parent.

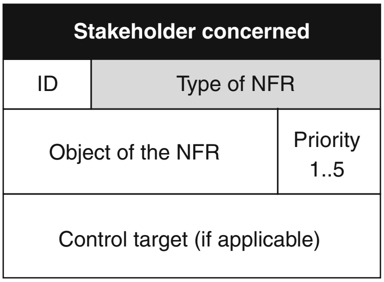


Fig. 3 Representation of a high-level NFR

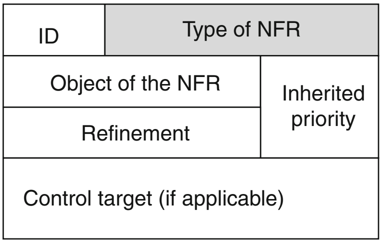


Fig. 4 Representation of a sub-softgoal

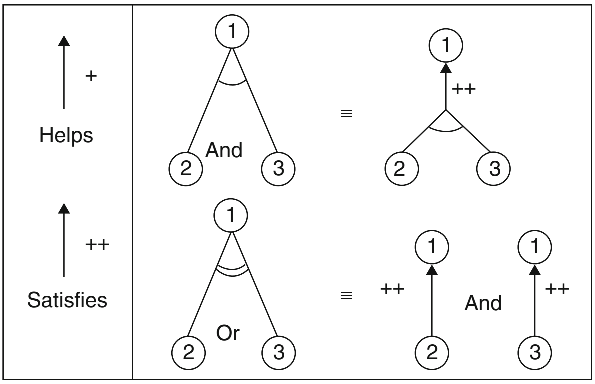


Fig. 5 List of available connectors used for softgoal decomposition

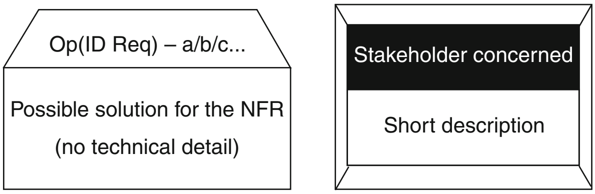


Fig. 6 Representation of an operationalization and FR(*right*)

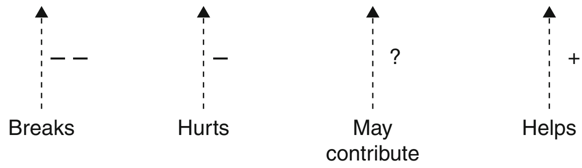
Sub-softgoals are subdivided until the analysts decide that no more decomposition is feasible or interesting. If possible, the analysts must try to avoid dividing a subgoal using only 'helps' contributions. If a softgoal does not have a sub-softgoal (or a group of sub-softgoals) which 'satisfies' it, it will be hard to determine if the higher level softgoal can actually be met. Once no more division is needed, possible operationalizations are associated to each leaf of the constructed graph. To do that, one must think about the most intelligent solutions to respond to the need expressed by a softgoal. All these alternatives will be represented on the graph using a unique ID (see Fig. 6) and linked to the corresponding softgoal using the same connectors than before.

At the end of this phase, the analyst should have a graph of formulated NFR, each with its own set of subsoftgoals and operationalizations. The next step is to study the interactions between graphs and to incorporate the necessary FR.

3.4 Studying Interdependencies in the Requirements Diagram

At this point of the process the analysts concentrates on the interdependencies between the different requirements. First all the NFRs graphs which are related are grouped. Obviously, the NFRs controlling the entire system must appear on each diagram. For the others it is quite easy: the NFRs linked to the same use cases, or having the same 'object' (Fig. 3) can interact or be in conflict. Once all the related NFRs are on the same diagram, one uses the use case diagram and the associated documentation to add all the related FRs to the diagram as shown on Fig. 6. FR can also have an ID.

Fig. 7 List of available connectors to study the interdependencies



In the case that different solutions have been considered to respond to a FR, these alternatives are represented in the diagram using the operationalization representation (Fig. 6) and links them to the FR according to the connectors of (Fig. 5). For each operationalization on the diagram, the systems analysts must study its interactions with the other elements of the graph (Formulated NFRs, sub-softgoals, FRs and other operationalizations) and insert links using the connectors given on Fig. 8. The 'May contributes' (?) link can be used during the process as a reminder to the systems analysts if he needs to ask for help to a stakeholder or a domain expert, but can't appear on the diagram at the end of the process.

Figure 7 depicts the refinement of the NFRs and operationalizations in a diagram. Within the operationalization there are two status indicators, the current and expected. The former is used for verification and is addressed at a later section.

What remains is the selection of the operationalizations which will be used to design the system. In this work, we don't propose any algorithm (this forms part of the future investigation on this topic) to solve this problem, but such an algorithm should:

- analyze the impact of choosing each operationalization (which requirements are satisfied or helped, which operationalizations or requirements are hurt or broken)
- assure that every FR (and NFR of priority 5) is satisfied (one of its operationalization is chosen)

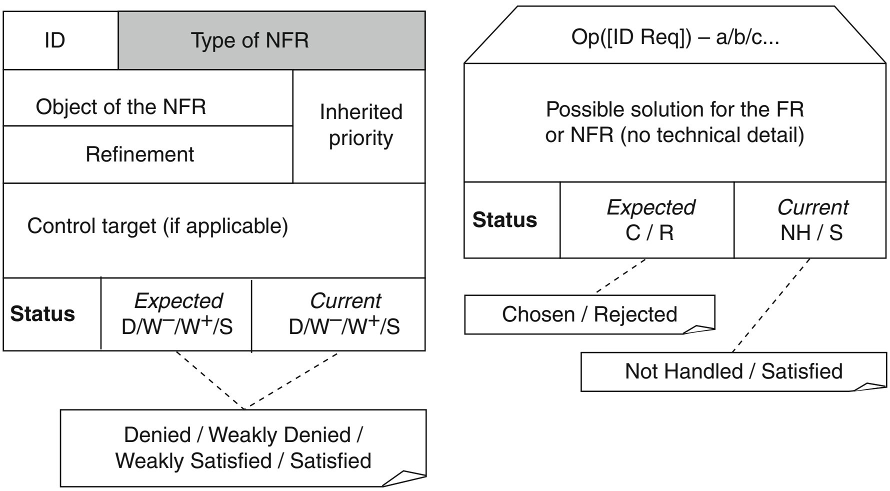


Fig. 8 Refinement of the representations

- give a mark to each alternative subset (using the weight of the contributions and the priority of the NFRs)
- return the highest-ranked solutions

The analyst then chooses the most appropriate subset for the design of the system. The chosen operationalizations is marked with the letter **"C"** in the 'Expected status' box, while the others with the letter "**R**", "Rejected". The 'Expected status' of the Formulated NFRs and softgoals will then be updated according to this choice.

3.5 UML Diagrams

This section demonstrates how to integrate NFRs and their operationalizations into UML diagrams. The focus is on sequences, deployment and classes but the techniques can work on other diagrams.

The following links can be used in these diagrams, as shown in Fig. 9:

- The «implements» link means that the element has been added to the design of the product in order to satisfy the operationalization. If the operationalization is not considered anymore, this element can be removed.
- The «contributes to» link means that an element previously existing in the diagram is used by the operationalization. It is important to highlight such an element: if it has to be modified to comply with its primary function, the systems analysts must study the impact of the change on the realization of the operationalization.
- The «controlled by» link defines that one knows that the implementation has to be guided by a particular NFR but no specific solution has been chosen yet. Most of the time, such a link must come with an annotation.

The general method to integrate NFRs and operationalizations into a diagram is by analysing which use cases (or which functional requirements) lead to representing this part of the diagram, what are the NFRs which control these use cases and then to study if the NFRs (or their operationalizations) are related to an entity in the considered sub-diagram. The following examples aim to make this process clear.

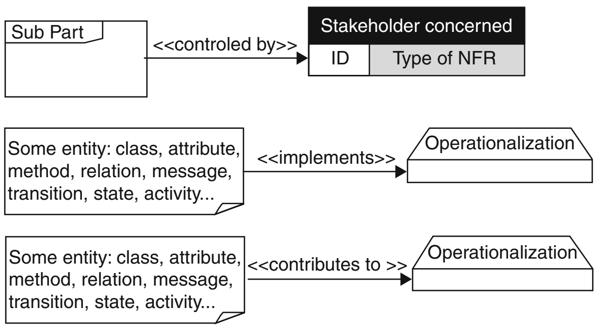


Fig. 9 Links to be used in UML diagrams

3.5.1 Sequence Diagram

Previous examples demonstrated how to integrate NFRs into use case diagrams. To incorporate the NFRs into the sequence diagram, the analysts will first construct the sequence diagram without taking into account the NFRs. This can be done using standard process, and the way it is done has strictly no impact on the rest of the process.

Once the sequence diagrams are ready, the analysts will incorporate the NFRs into them. Usually, a sequence diagram will be related to one of the use cases that has been design earlier, and this use case might be related to one or more NFRs. If this is the case, then for each of them, the analyst has to ask himself: where does this NFR impact the sequence diagram, and how? The analyst can choose the level of detail for the diagram as shown on Fig. 10. Therefore the analyst can:

- state that a subsequence is «controlled by» an NFR,
- link some specific previously existing messages to a operationalization («contributes to»),
- add new messages (and maybe new classes) to the diagram which «implements» an operationalization.

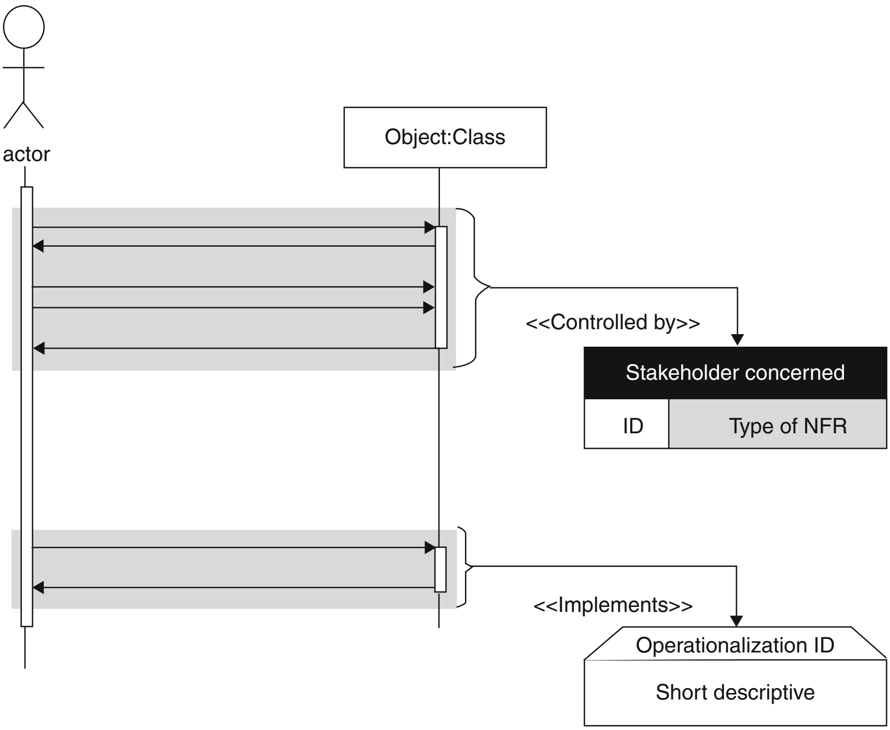


Fig. 10 Representation of an operationalization

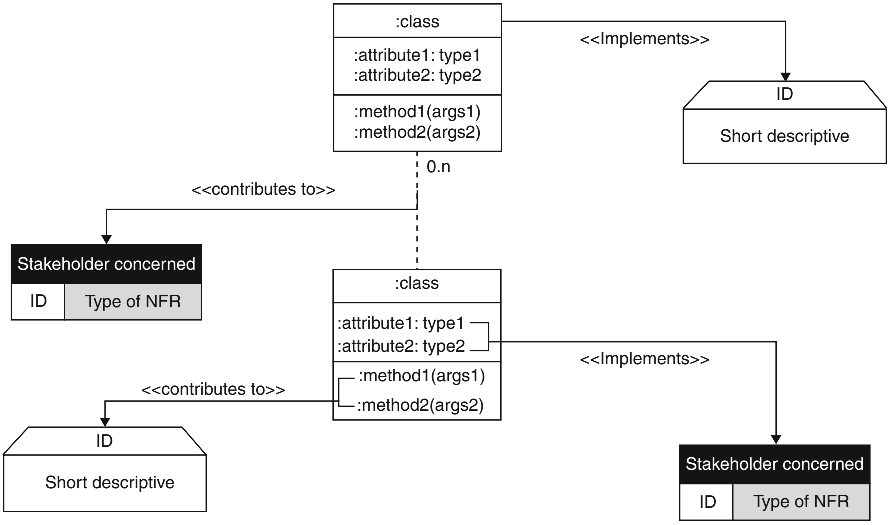


Fig. 11 Incorporating NFRs into classes diagrams

3.5.2 Class Diagram

When it comes to class diagrams four different entities are of importance: classes, attributes, methods, and the links between them. All of these entities can implement or contribute to an operationalization, or be controlled by a NFR or one of its subgoals. In the diagram of Fig. 11, it is very important to provide the designers with an easy way to find which elements help with satisficing the non functional requirements. For instance, in the case of an attribute, if it implements an operationalization, then it means that it is not directly linked to the FRs of the system. If it is not properly referenced as being linked to an operationalization of the NFRs, the designers would have no way to know why this attribute exists in the future. Similarly, if our attribute contributes to an operationalization or is controlled by a NFR, we do not want it to be modified. This could happen if the attribute is not clearly linked to a NFR or an operationalization. Similarly, this reasoning can be applied to classes, links between classes or methods.

The approach to incorporate NFRs into class diagram is quite simple. When the analyst constructs the diagram, usually he looks at sequence diagrams or use cases, and implements the relevant entities. Since the NFRs or their operationalizations are clearly stated in these diagrams, it is easy for the analyst to see which NFRs might have an impact on the class (or other entity) and accordingly incorporate the NFRs in the diagram. After this process, all NFRs will have been taken into account, except the global NFRs. This is because they are not linked to any use case, neither stated in the sequence diagrams. At this time, the analyst has to refine the class diagram to include them.

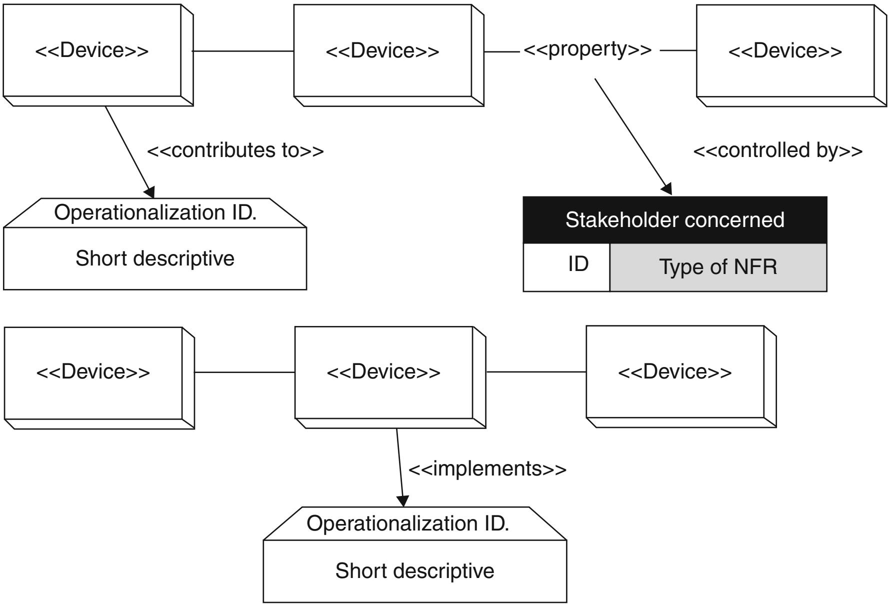


Fig. 12 Incorporating NFRs into deployment diagrams

3.5.3 Deployment Diagram

At this point of the process, the analyst has a good understanding of the system. Deployment diagrams describe the interactions between the system being designed with other systems. NFRs of course apply to such interaction, the most obvious ones being security, reliability and performance. This is very important, because many NFRs cannot be satisficed by implementing new classes or attributes in class diagrams. An NFR such as 'The search must take less than 5 s' might be overlooked if we do not consider the deployment diagram. Figure 12 shows how to link NFR and operationalizations to elements of the deployment diagram.

3.6 Verification

Once the NFRs or their operationalizations have been included in every necessary diagram, one has to verify that everything planned on the NFR graphs is actually considered in the design of the product. This justifies the esistance of the box 'Current status' at the bottom of the representations of NFRs and operationalizations in the NFR graphs. For each operationalization chosen, one has to check if it appears on diagrams with links of type «implements» or «contributes to». If it is the case, then the operationalization has been 'satisfied', else it was 'not handle'. Then update the 'current status' of all NFR softgoals that the operationalization links to.

After considering the operationalization, if an NFR softgoal that was expected to be 'satisfied' or 'weakly satisfied' is not, it has to be verified if this NFR is linked to any other diagram with relation «controlled by». If so, the NFR has not been forgotten, however, no technical solution has been indicated on the design. If it is the case, it is up to the analyst to decide if this is good enough to mark to softgoal as 'weakly satisfied'/'satisfied' or if the design needs to be refined. These verifications should in fact be done automatically by a framework implementing our proposal. After these verifications, the operationalizations which were chosen but are not satisfied and the NFR softgoals where 'expected status' is different from 'current status' should be highlighted for the analyst to know immediately what is missing.

3.7 Case Study

This cases study demonstrates the application of the method for designing an e-commerce application for online car rentals. Due to space limitations, the example concentrates on core aspects of such a system with emphasi on business processes

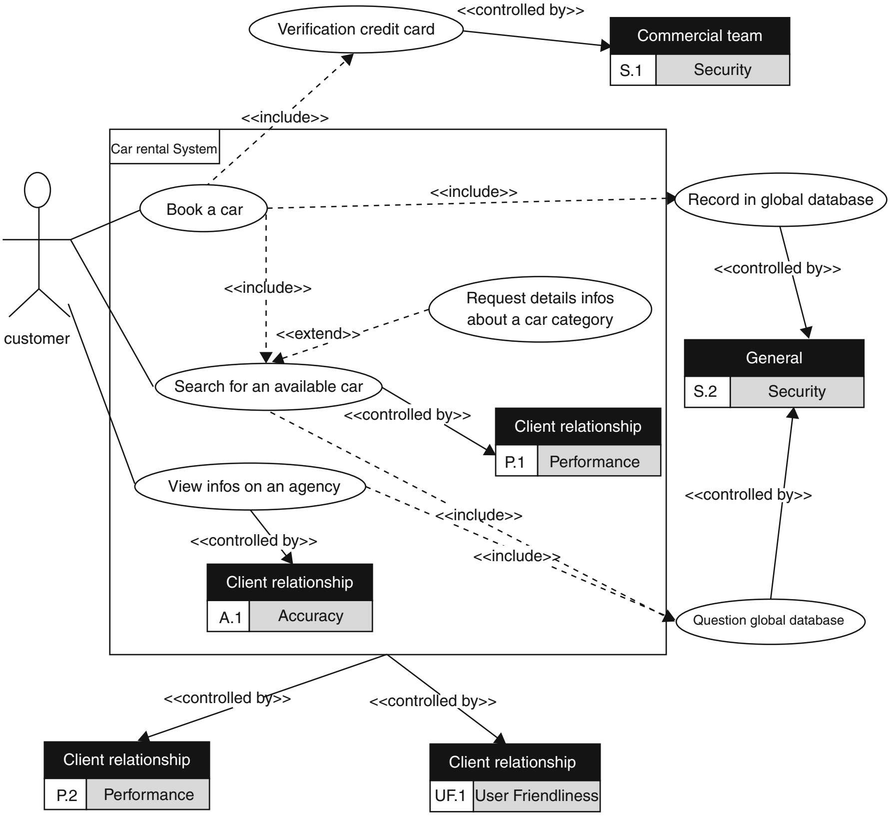


Fig. 13 Use cases enriched with NFRs

that relate to rental and provision of information to customers regarding the company. Therefore, the following business needs are addressed:

- The user can get information about the local agencies in which he will pickup the vehicle. Related information include: the address of the agency and its opening hours.
- The user can search in a database for information about the cars.

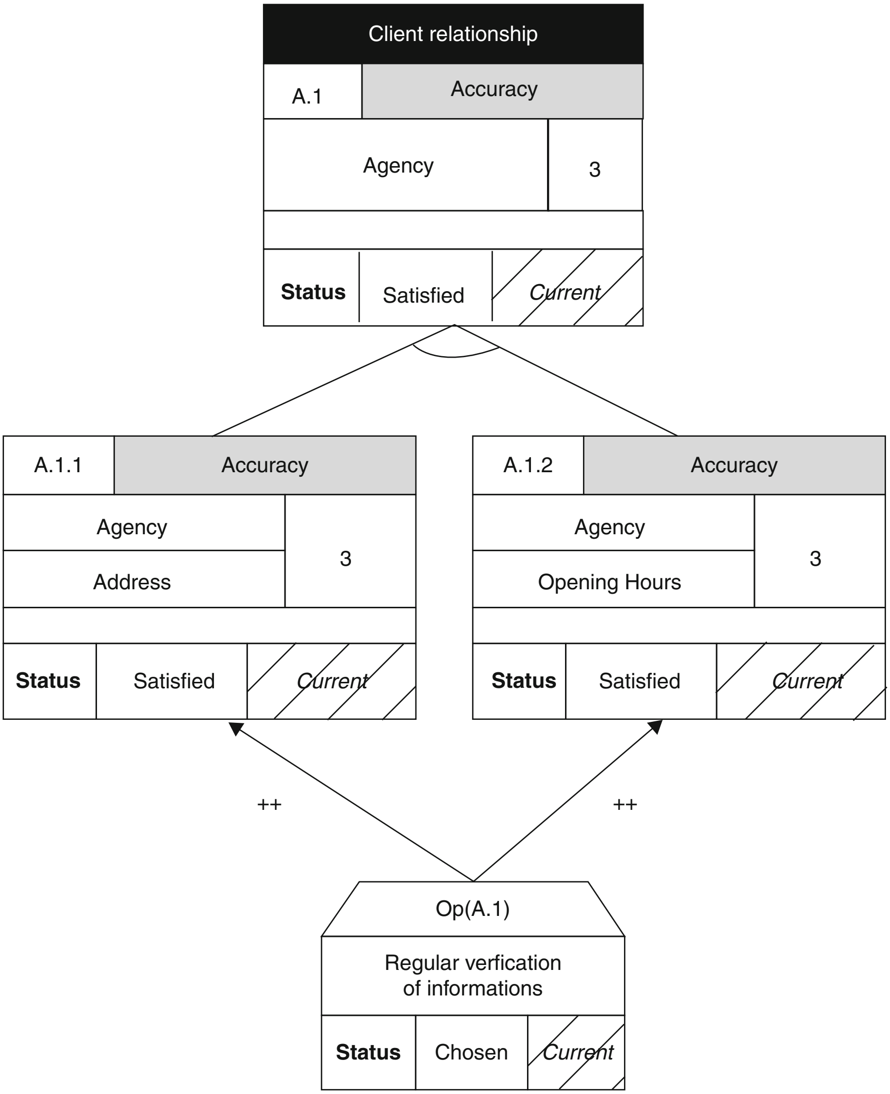


Fig. 14 Requirement diagram relating to the information about local agencies

- The user can request a quote for a car.
- The user can book a car on the website and pay online for the booking.
- The site must include some advertisements.
- The non-functional requirements that we consider are the following:
- The information about the agencies should be up to date.
- The security is considered vital. This includes the connection with the bank (for the booking) and the connection to the car database.
- The search facility should be efficient
- The web site should be user friendly
- The pages should load quite fast, so that users won't get annoyed the whole process.

Since the method focuses in the integration of NFRs into UML diagrams an the drawing of NFR graphs, the only FR related elements included are those impacting on NFRs. From these requirements, we can draw the diagram on Fig. 13.

The next step of the method is to draw the requirement diagrams. We know that we will have to refine some of our requirements and to choose which ones will be included in the design of the website. For example, an attractive website includes flash animation, but this is in conflict with the NFR that relates to the loading time of pages. This needs a trade off, and we consider that users will not want a slow navigation just to have fancy animations on the website. Two different diagrams (Figs. 14 and 15) are drawn, since this study could not have been demonstrated through one diagram. This is performed without splitting requirements that impact on each other.

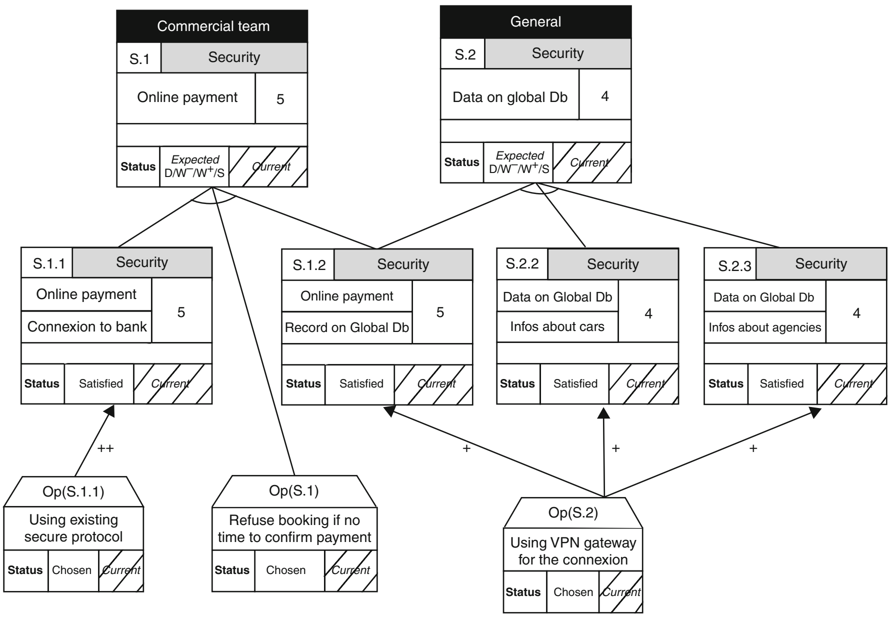


Fig. 15 Security requirements of the system

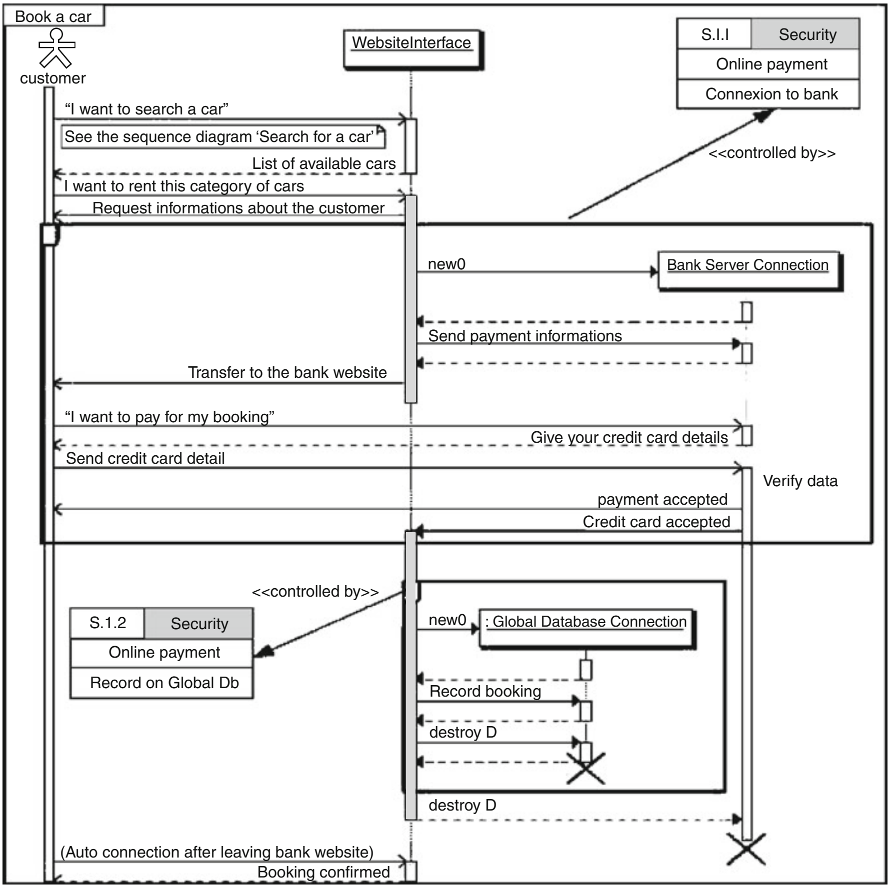


Fig. 16 Sequence diagram of the booking process

The sequence diagram of the system is depicted in Fig. 16. The functionality depicted is limited to the car booking process. From the use case, we can see that this diagram will be related to two different NFRs, both of type security. First, the connection with the bank needs to comply with a security NFR, which is clearly stated in the sequence diagram. Later on in the process, we will record the transaction in the global database, which is also impacted by a security NFR.

4 Conclusions

This paper presented a novel approach to dealing with Non-Functional Requirements and to model them in UML diagrams. This work constitutes an important contribution given that it allows system analysts to consider NFRs in their models without requiring additional learning of new software. Part our future plans include the automation of elements of the proposal in order to utilise its full usefulness in industrial settings. We accept that the full proposal is very difficult to automate, as there is a need to provide manual support by systems analysts at certain steps of the approach. However, this is the case with almost all methods proposed so far in this area.

References

Chung L, Nixon BA, Yu E (1995) Using non-functional requirements to systematically support change. In: ‘RE ’95: proceedings of the second IEEE international symposium on requirements engineering’, IEEE Computer Society, Washington, DC, USA, pp 132–139

Chung L, Nixon BA, Yu E, Mylopoulos J (2000) Non-functional requirements in software engineering. Kluwer Academic Publishers, Boston

Cysneiros LM, do Prado Leite JCS (2001) Using uml to reflect non-functional requirements. In: ‘CASCON ’01: proceedings of the 2001 conference of the Centre for Advanced Studies on Collaborative research’, IBM Press

Cysneiros LM, do Prado Leite JCS, de Melo Sabat Neto J (2001) A framework for integrating non-functional requirements into conceptual models. Req Eng 6(2):97–115

Finkelstein A, Dowell J (1996) A comedy of errors: the London ambulance service case study. In: ‘IWSSD ’96: proceedings of the 8th international workshop on software specification and design’, IEEE Computer Society, Washington, DC, USA

OMG (2007a) Systems Modeling Language (SysML), V1.0. http://www.omg.org/spec/SysML/1.0/PDF. Accessed 1 July 2011

OMG (2007b) Unified modeling language (UML), v2.1.1. http://www.omg.org/spec/UML/2.1.1/. Accessed 1 July 2011

Russell M (2004) Quality busters: forget the environment. http://www.ibm.com/developerworks/web/library/wa-qualbust1/. Accessed 1 July 2011

Smith LW (2000) Project clarity through stakeholder analysis. CROSSTALK J Defense Soft Eng 13(12):4–9

Zou J, Pavlovski CJ (2006) Modeling architectural non functional requirements: from use case to control case. In: ‘ICEBE’, IEEE Computer Society, pp 315–322

Bottleneck of the Development of Public Information Systems

Jaroslav Král and Michal Žemlička

Abstract It has been assumed that information technologies and public information systems like e-government and e-health would bring substantial positive effects. The current results of public information systems are not too optimistic as there are barriers due to an improper setting of legislative and related processes, especially of the (personal) data security rules and processes. The processes in fact almost completely exclude the computation of open information from sensitive data. We show that it has substantial negative effects but it still does not assure required personal data security. The undesirable effects are not only important, but sometimes they are fatal. This problem is frequently underestimated and often overlooked. The current practices substantially limit the applicability of the concepts of artificial intelligence, knowledge society, and semantic web. We propose solutions enabling to change current undesirable practices like massive data deletion. The solutions are based on the combination of processes performed by a trusted body. We discuss technical solutions enabling implementation of our proposals and specifying ways of changing current legislative to enable them. It must be based on the change of social and legislative conditions like prejudices of public (Big Brother hysteria).

1 Introduction

The progress of IT is impressive. Hardware systems and especially software systems offer more and more power as well as principally new solutions. Software passes through a revolution every 10 years (global networks, web, SOA, data mining, semantic web, etc.).

One can expect that it contributes to an improved enterprise management as well as to an improved government of states. Whereas the better management of enterprises is to a high degree a reality, the improvement of the government (macroeconomic)

J. Král (✉)
Charles University, Malostranské nàm. 25, 118 00, Prague, Czech Republic
e-mail: kral@ksi.mff.cuni.cz

M. Žemlička
e-mail: zemlicka@ksi.mff.cuni.cz

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_30,

processes is doubtful, see e.g. the current economic crisis or the ill conceived changes in education.

The power and effects of e-government and e-heath and other public information systems could be substantially enhanced if the sensitive data security rules and practices were changed.

The current results of public information systems are not too optimistic as there are barriers due to an improper setting of legislative, especially of the rules and processes of (personal) data security. The processes in fact ban the computation of open information from sensitive data. We analyze the substantial negative effects of this practice. We show using case studies that the important undesirable effects are very important but they are underestimated and often overlooked. The current practices substantially limit the applicability of the concepts of artificial intelligence, knowledge society, and semantic web.

We formulate proposals how to change the current undesirable practices including such brute measures like massive data deletion. We discuss technical solutions enabling implementation of our proposals and indicate how to change current legislative to enable them.

2 Crucial Data Security Issues

We believe that one, may be the most important, reason of the data security issues is the practice how e-government systems collect and use economic and social data. The analysis (see below for details) of the rules and practices of government processes indicate that the processes cannot be properly applied as the data (or information) needed for decisions of majorities, enterprises, and individuals are not accessible even if the needed information should be open.

Any information computed from sensitive data is in fact by current practices treated as sensitive. It is the sensitive data cannot be used to produce any information being public, i.e. the information that should be accessible for all citizens.

There are ill-conceived rules based on the requirement that data should always be used only for purposes for which they were collected and only by the bodies allowed doing it. So nobody is often allowed to compute mean age of people living in some region or to perform sociologic studies. This limitation is somewhat meaningful as it blocks some data misuse in marketing. We are convinced that such a rule is too strong and should be weakened; for example, statistical analysis should be allowed.

The consequences are fatal. For example, it is impossible to have independent control of the outputs of rating agencies or to enable people to evaluate the quality of schools. It is also the barrier of long term (strategic) planning of enterprises.

The information needed for the evaluation of education is individual for evaluator. It can be, however, anonymous with respect to persons.

As such it should be accessible to any application if there is an application able to compute it. The application must use personal data.

The solution is that the application must either use data transformed to be anonymous or the application is executed by an accredited (trusted) institution. Both measures should be combined to achieve satisfactory level of private data security.

The security can be further enhanced by filtering of output information that could disclose some private data (e.g. to identify an individual). The filtering should be also performed by the accredited institution. The institution should therefore perform a collection of processes like non-standard data anonymization, intelligent data and information searching and filtering.

The proposed solutions are based on the orchestrated collection of individual actions that could seem simple and straightforward. The orchestration is, however, difficult as it must coordinate technical as well as nontechnical aspects. The solution in fact depends on the coordination of software vendors. It is the reason of limited success and contribution of public information systems. Some of the reasons are described below in details.

As the output information can be occasionally sensitive depending on the structure of data, i.e. when the output occasionally regards one person only, the outputs must therefore often be filtered.

3 Brute Data Security Measures in Action

Hysteria about Big Brother and the common (mis)interpretation of Universal Declaration of Human Rights leads to the preference of brute action based on data deletion/destruction. It can have important adverse effects. Let us give an example.

Some people in Czech Republic illegally produce narcotic Pervitin from medicaments containing pseudoffedrine. The medicaments can be bought freely. In order to prevent it an on-line system monitoring the purchases of all medicaments in pharmacies was built. Pharmacies were obliged not to purchase the critical medicaments if a buyer had bought suspiciously large amount of the medicaments lately. It is, in several weeks.

A system S monitoring all drug purchases all over Czech pharmacies was developed and implemented. The result was steep decrease of the Pervitin production. The system was banned. The root cause of the problem was the unbalance between data security requirements (i.e. the prevention of data misuse) and the duty to provide any open information computable from existing data. We will show that a satisfactory equilibrium between data security and information accessibility can be found. The solution is based on principles and techniques discussed below. After a short time of the operation of the system S the Office for Personal Data Protection banned the use of S and forced deletion of the collected data. It has the following effects:

1. A steep increase of the production of the narcotic Pervitin and of related criminal activities.
2. The on-line supervision of medication was blocked. The supervision was able to avoid many erroneous medications and treatments, i.e. the ones being in conflict with some other running ones. Such cases can happen due to missing information (e.g. after health incidents) or due to occasional mistakes of physicians having a limited access to the health picture of the patient. It can avoid the occasional but not rare health damages, fatal cases inclusive.
3. Avoidance of processes allowing optimization of health-care expenses and even to improve health care.

4. Data for health care (medical research) or operative actions during epidemics were blocked.

We can conclude that such a practice declared to protect human rights in fact threats them. They may threat life or health and consume means provided by citizens to health care system.

4 What Data Security Level is Reasonable and What is Ever Achievable?

Our proposal requires that the data needed to compute open information are optionally partly anonymized, partly copied into a secure place, if necessary, and accessed by accredited applications computing the needed (public) information. The applications can be provided by people or institutions needing the information and accredited by a corresponding accreditation body.

It follows that the data can be (are) less secure than in the case when almost no information producing applications exist. It is often assumed that it is absolutely not acceptable.

This requirement is too strong. We must take into account that people and enterprises provide their sensitive data to many subjects. Examples are:

- financial institutions (they are quite secure),
- social software systems (blogs, Facebook, ...) are insecure, even if one is careful (what is not usually the case),
- data on business partners,
- personal contacts on private data stores,

All these cases can cause substantial data leakage. There are yet another ways of data leak:

1. Mobile phones can be wiretapped and, moreover, the mobile devices can disclose their actual positions. The mobiles can be monitored from satellites
2. Communication over internet is not encoded enough.
3. There are private data being open by law (like land register, data on club/society members, or registry of enterprises).
4. Many servers storing/handling personal data are only partly immune against hacker attacks.

Under these circumstances our proposal, if implemented properly, practically does not decrease the data security as also after introduction of the new system the main data security issues will be the ones listed above.

5 Data Protection and the Evaluation of Education

We will discuss the design principles of our solution on the case of the evaluation of quality of education. The quality of education is the crucial element of the long-term prosperity of individuals as well as the entire state or society. The existing data can be used to make the evaluation of education less subjective—not based on rumors or gossips or on not representative or inflexible analysis of the sample research.

The result is that there are different judgments on the effect of the current education. For example, the STEM (science, technology, engineering, and mathematics) oriented education in USA is considered to be inefficient [2, 4, 8–10, 12]. Similar complaints were in fifties [12]. The researches of personal agency Manpower [7] indicate the lack of STEM-based professions.

What are the reasons for it? There were complains on STEM education in the USA in fifties but the science and industry in the USA are still the world leaders. Is it durable? What are the reasons of current situation? Are STEM jobs well awarded? We do not know. Not only in the USA

The complaints on the falling quality of STEM education exist in the Czech Republic where the situation in education is complicated by turbulent school reforms. The reforms are supposed to be the reason for the falling quality of general education—not only of the STEM one.

The quality of education is for the majority of people correlated with career success of graduates and alumni. The success is measured (judged) by employment level, salaries, and job prestige. It follows that the answers must be in some sense necessarily obsolete—we estimate the success of current students by the success of past graduates/alumni. This limitation is principal.

6 The Quality Concept

We must understand that quality is a multidisciplinary concept and that it depends according ISO 9000:2005-Plain English Dictionary on the interests and aims of the quality evaluator. The evaluator can be an individual (e.g. a parent) or an institution.

It follows that different school rankings are of limited use as they need not reflect real needs of individual citizens or institutions. The system should use data on labor market. Such data are usually sensitive, often personal.

What is the solution? It is necessary to enable the citizens or institutions to generate their own ad-hoc "quality rankings" using queries over data. The queries can be based on criteria felt to be relevant by a given evaluator. It is technically feasible but we must invent processes and techniques assuring a good level of data security.

Our solution can be generalized in the following way: It can be possible in the case of nonstandard data analysis to develop an application implementing the analysis. The application can be after successful accreditation integrated it into the evaluation system. The use of the application must not spoil data security.

7 Data Security and Measuring of Economic Processes

The current economic depression has shown, that the feedbacks in global economy are unreliable and slow. The forecasts of economic developments are made almost exclusively by analytical bodies that are allowed to access sensitive data and that often must do sample researches being slow and subject of statistical errors. The analyses are not always free of subjective judgments.

The bodies are looking for answers to the "most important" questions that need not be of primary interest for a given subject. The outputs of the bodies are difficult, if not impossible, to be verified or tested independently.

Such issues could be solved if all important economic data, the sensitive ones inclusive, were collected, filtered, and cleansed by an accredited body and made accessible to public for the computation of open information in the way described below. Such a solution is not desirable for powerful lobbies of subjects having advantages from the current situation (having exclusive access to data or performing sample researches).

8 Principles of Implementation of Systems Providing Open Information from Sensitive Data

It is right not to allow open access to personal data—even in the case that the data are anonymized. It is very likely that the anonymization is not powerful enough or it is too restrictive. We therefore propose a system having no such drawbacks.

The system is used by a trusted (accredited) body (AB) for the computing of open information from personal data. It can have the following structure:

1. *Data input.* The data must be looked for, filtered and cleansed. The data should n stored in a dedicated data store of the trusted body. The on-line distributed data access is also possible. Both solutions have their cons and pros, the dedicated data store is usually preferable. Both solutions can be combined. They almost fully exclude the use of open tools like web search engines.
2. *Optional data anonymization.* The data can be partly encoded, e.g. personal IDs can be encoded using asymmetric encoding. Such a measure is good to prevent hacker attacks. The trusted body should be able to decode the data if necessary.
3. *Generation and filtering of output information.* This function is necessary to prevent occasional undesirable sensitive information leakage like in the case when given piece of information identifies a too small group of individuals. This problem practically excludes the public accessibility of anonymized data.
4. *The open information* being output of the system can be stored in open data stores and can be looked by web indexing robots.

The implementation can be designed in the following way (see Fig. 1):

- The data are organized into a (virtual) secure data store SDS. The data store can include (as services in the sense of SOA) the data tiers of data providers. Other data providers can provide the data in batch (bulk) mode.
- SDS is maintained and controlled by an accredited body.
- The outputs of SDS are controlled (approved) by the accredited body AB using some automated tools.

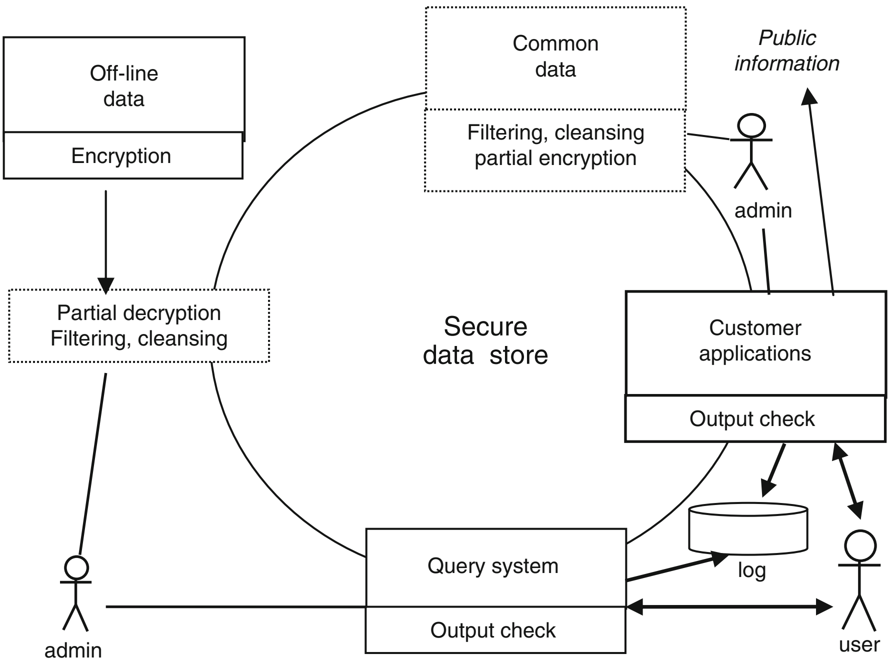


Fig. 1 The structure of the information system enabling to use sensitive data for production of nonsensitive information. Partial encryption or decryption is used as a sophisticated anonymization procedure

- The outputs are logged for possible after the fact analysis of outputs misuse. We must use such a measure as the output checks can be very effective. They, however, cannot be perfect. The users must, moreover, agree that the use of the information being not fully anonymous is prohibited.
- The outputs can be produced by a flexible database system using general queries covering the needs of typical users. It is possible that advanced users write an application providing a more sophisticated information. Such application must be accredited by AB and integrated into the system. The AB can supervise the outputs, if desirable.
- The input data must be cleansed and filtered and their sources must be accredited by AB.

We believe that the principles enable a satisfactory data security level. The problem is that a too high security level is required now. The existing security level implies data use barriers having almost no positive effects but many negative ones.

9 Politics of the Use of Personal Data

Let us discuss the ways of the implementation of information producing applications in details. The information that can be mined can be of very different types that cannot be predicted as information needs vary. The needs depend on the interests of users (groups of users) and even on data and information already owned by the users.

It is therefore no good requirement that all information producing applications should provide the developer of public information systems. The number of information types is too large, the needs vary, and any proper solution should be flexible. Moreover, if a proper service-oriented architecture is applied (and it should be so [5]) then many developers can provide the applications in a seamless way.

So the best solution is to allow or use many autonomously developed information producing applications, typically developed by users themselves. It implies that public information systems should be flexible systems able to integrate any number of information producing applications certified by a proper governmental body. Such requirement can be easily met if the information producing applications are integrated as software services provided that the information systems have a serviceoriented architecture. The integration of information producing applications is then the same as the integration of information systems of individual governmental offices into the software supporting *e*-government.

As it was shown above, the information producing applications can be agreed (accredited) for integration by an authorization body. The checking (filtering) of the information producing applications properties can be adapted to the level corresponding to the authorization of the author of the information producing application and according to the requirement of the information producing application on data input (public data input imply almost no checking-input command only need to be checked/evaluated).

An information producing application can have (but need not have) the form of a web service (compare [1, 13, 14]).

10 Neglected Data Quality Issues

The discussed problems of personal data security can be treated or viewed as a process of destroying of one aspect of data quality [11], it is the data accessibility.

The issue is a part of a broader problem of the treatment of data quality protection in public information systems, especially in e-government. The poor data quality can be the consequence of neglecting the problem of data saving and use. It is in turn the consequence of legislative gap of current legislative in many countries does not require or state any processes assuring the preservation of main data quality aspects, especially timeliness, completeness, and consistency based on well-formed processes of data acquisition and compatible data formats. The data are then less useful and they do not enable many processes generating crucial information like indication of coming economic depression.

The main reason of such a situation is the missing awareness of the society, especially of the government, that the situation has very important negative consequences. Then therefore is no substantial pressure to change it. It holds in spite of the fact that the poor data quality implies substantial economic losses and in emergency situation can cause life losses. In long terms it can cause the deterioration of the services of crucial institutions.

We believe that IT experts should try to change the situation. It is good for the society as a whole and especially for the further development of IT.

11 Conclusion

The current rules and practices of data security are too limiting. They limit the access to information that should be public. The limitation is not caused by technical barriers. We have shown that it is technically feasible though not easy to provide access to all information that should be public.

The main issue is that societies (states), not to speak about individuals, are not aware of substantial direct and indirect costs of current data practices. It is yet more important that the practices have crucial negative long-term consequences.

We have discussed the consequences in the case of education systems being now in fact dynamic systems without proper feedback. The current practice of evaluation/computation of macroeconomic data/information implies that national/global economies are also dynamic systems without satisfactory feedback. Such systems tend to be unstable. It can be one of the reasons of the current global financial and economic crisis.

Note further that the practices limit the effectiveness of estimations provided by rating agencies and blocks the possibilities to make independent ratings and economic research based on sufficient data.

The current treatment of health data reduces the efficiency and increases the costs of medical procedures as some health-care data (personal anamnesis inclusive) are not accessible.

We believe that the above discussed proposals are the necessary precondition of the improvement of current situation. The issue is not easy to solve from the technical point of view, but it is solvable.

The crucial barrier is in the legislative area. The legislative limits the access to data but it does not procure any institutions and procedures enabling access in principle to all publishable information. The principles of possible solution were discussed above.

The crucial problem is that information system development must not take the existing environment (society prejudices, legislative barriers) as given. We must show that challenges and threats of current practices. In other words, we must be more clearly involved on social processes. The treats are not the treats of the users only. It is the bottleneck of public information systems of the sense of [3]. If the bottleneck is not changed, no substantial enhancements of the effects of public information system can be expected. We cannot expect that any knowledge society can be built [6].

Note that current situation can be compared with an enterprise having a powerful business intelligence system. The enterprise, however, does not allow its staff to use it. Such an enterprise will probably fail. It is not excluded that in similar situation the nations will fail in long terms too. It can be concluded that IT experts are obliged to fight against the prejudices and procedures spoiling the perspectives of nations in general and positive effects of IT in particular.

Acknowledgements This work has been partially supported by the Czech Science Foundation by the grant number 201/09/0983.

References

1. Ang J, Cherbakov L, Ibrahim M (2005) SOA antipatterns. http://www-128.ibm.com/developerworks/webservices/library/ws-antipatterns/
2. Committee on Engineering Education, National Academy of Engineering (2007) Education of engineer of 2020. The National Academies Press, Washington
3. Goldratt EM (1997) Critical chain. North River Press, Great Barrington
4. Jamieson L (2007) Engineering education in a changing world. The Bridge Spring 2007. http://www.designcon.com/infovault/leah_jamieson_keynote.pdf
5. Kral J, Zemlicka M (2001) Electronic government and software confederations. In: Tjoa AM, Wagner RR (eds) Twelfth international workshop on database and experts system application. IEEE Computer Society, Los Alamitos, CA, USA, pp 383–387. doi:10.1109/DEXA.2001.953091
6. Kral J, Zemlicka M (2008) Bottleneck of knowledge society. In: Lytras MD, Carroll JM, Damiani E, Tennyson RD, Avison D, Vossen G, De Pablos PO (eds) The open knowlege society: a computer science and information systems manifesto. Communications in computer and information science, vol 19, pp 83–91. Springer, Berlin, Heidelberg. doi:10.1007/978-3-540-87783-7_11
7. Manpower (2008) Talent shortage survey: 2007 global results. http://www.manpower.com/research/research.cfm
8. Mervis J (2007) Congress passes massive measure to support reearch and education. Science 317:736–737. doi:10.1126/science.317.5839.736
9. Mervis J (2007) U.S. says no to next global test of advanced math, science students. Science 317:1851. doi:10.1126/science.317.5846.1851
10. Mervis J (2008) A new bottom line for school science. Science 319:1030–1033. doi:10.1126/science.319.5866.1030
11. Pipino L, Lee YW, Wang RY (2002) Data quality assessment. Commun ACM 45(4):211–218. doi:10.1145/505248.5060010
12. Spellings M (2006) Answering the challenge of a changing world. U.S. Department of Education. http://www.ed.gov/about/inits/ed/competitiveness/strengthening/strengthening.pdf
13. W3 Consortium (2000) Simple object access protocol. http://www.w3.org/TR/SOAP
14. W3 Consortium (2002) Web services activity. http://www.w3.org/2002/ws/

A Framework for Optimizing Inter-operating Business Process Portfolio

Tri A. Kurniawan, Aditya K. Ghose and Lam-Son Lê

Abstract A company or an organization may have a number of business processes depending on its size, represented in a particular business process modeling language such as BPMN. They may have inter-operating relationships, i.e. one process may depend on other processes by exchanging messages in order to achieve its objective(s). One of the process improvements is to optimize its processing time, e.g. by parallelizing some tasks within a particular process. Optimizing processes individually without considering any inter-operating relationship between them will potentially produce a process which is not consistent with the original one w.r.t. the pre-defined business rules. This paper introduces a novel approach of optimizing process portfolio both locally and globally by considering inter-operating relationships between process models as well as taking into account the minimal change strategy.

1 Introduction

A business process consists of a set of activities that are performed in coordination in an organizational and technical environment [1]. A process is thus a specific ordering of work activities across time and place, with a beginning, an end, and clearly defined inputs and outputs: a structure for action. A company would have a number of business processes depending on its size. There could be relationship or dependency among the processes. Many process designers may develop and maintain a process as a single entity without considering the others which may be in relation. Such a consolidating view of the complete landscape of business processes is the

T. A. Kurniawan (✉) · A. K. Ghose · L.-S. Lê
Decision Systems Lab., School of Computer Science and Software Engineering
University of Wollongong, Wollongong, NSW 2522, Australia
e-mail: tak976@uow.edu.au

(On leave from a lecturership at University of Brawijaya, Indonesia)

A. K. Ghose
e-mail: aditya@uow.edu.au

L.-S. Lê
e-mail: lle@uow.edu.au

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_31, © Springer Science+Business Media, LLC 2011

heart of process portfolio management, a coherent treatment of the entire set of processes, allowing them to be improved in total, rather than streamlining one and, consequently, unknowingly, sub optimizing others [2].

Process portfolio optimization is an important part in process portfolio management in order to perform a coherent treatment of the entire set of processes for improvement. Process improvement relies on the nature of change to business process. In some cases, the designer may perform some changes on a particular process model without considering any rules that may exist according to any relationships, such as inter-operating, to other process models. The changed process may structurally be correct to achieve the improvement objective but an inconsistency might be introduced in business rules. Instead of modifying the business process individually, process portfolio optimization would consider inter-operating relationship between process models being modified in order to meet both the improvement objective(s) and all pre-defined business rules (i.e. constraints that exist within a particular inter-operating business process, not bodies of the regulatory requirements).

We consider an inter-operating business process model as a process model which involves at least two processes represented in two pools and there exists at least one message exchanged between a task within the first process to one task within the second process. The inter-operating relationship could be either within an enterprise or between multiple enterprises. An inter-operating process exists within an enterprise when the involved pools represent roles (e.g. department, unit, etc.) in the enterprise. An inter-operating process exists in an inter-enterprise context if exchanging messages involves a number of pools from different enterprises.

The rest of this paper is organized as follows. Section 2 provides a running example that is used throughout the paper. Section 3 briefly discusses the related work. Section 4 outlines our proposed framework including all definitions and procedures. Section 5 illustrates our proposed framework based on the original process introduced in Sect. 2. Section 6 draws some conclusions and points out future work.

2 Motivation

In this section, we will illustrate an example of process optimization on an inter-operating business process involving two roles: a *Customer* and a *Supplier*. The *Customer* places an order to the *Supplier* and then the *Supplier* processes it. Once the *Customer* makes payment, the *Supplier* delivers the goods which was ordered by the *Customer*. This process is represented in Business Process Model and Notation (BPMN)[1] as shown in Fig. 1. Note that this process model has been enriched with immediate effect [3] (i.e. effect of an executed task that we will discuss later on) and the execution time of tasks. Let e_i represent immediate effect of task t_i, and estimated task execution time in days, denoted in d. This estimated time represents the required time in order to accomplish a task. For example, task *Make Payment* needs a total of 10 days including preparation in documents, bank account required, etc; not only

[1] BPMN homepage http://www.bpmn.org/

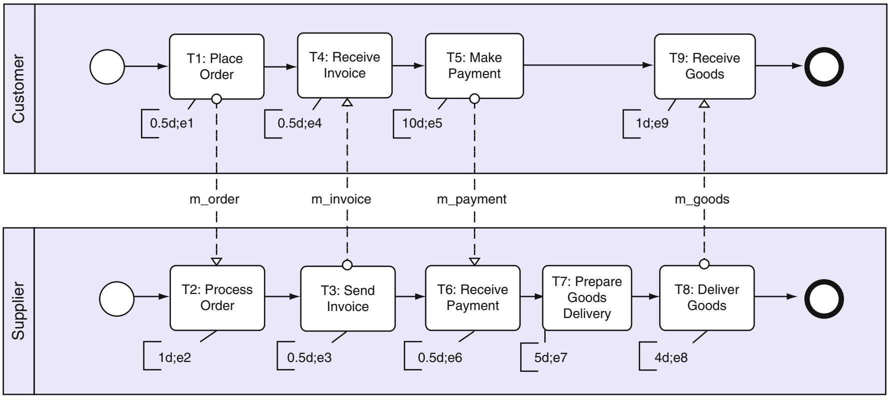


Fig. 1 The original process model of inter-operating *Order Processing* involving two roles: *Customer* and *Supplier*. Each task is annotated with *immediate effect* e_i and estimated task execution time in days *d*

transferring the money itself. We number each task with *Ti* (i.e. task i*th)* and will recall each task by its number in the following figures.

The process is performed in the following sequence: *Place Order, Process Order, Send Invoice, Receive Invoice, Make Payment, Receive Payment, Prepare Goods Delivery, Deliver Goods,* and *Receive Goods*. The sequence of activities must follow the order of the corresponding messages as a pre-defined rule, i.e. *m_order, m_invoice, m_payment* and *m_goods* such that *m_order* $\prec$ *m_invoice* $\prec$ *m_payment* $\prec$ *m_goods*. The constrained sequence means that message *m_order* precedes *m_invoice*, *m_invoice* precedes *m_payment* and *m_payment* precedes *m_goods*.

Due to a particular improvement on processing time, the designer decided to implement process changes by parallelizing tasks as much as possible on each process. In general, the designers pay attention on each process individually to get the most optimum design (i.e. a process model which has the shortest processing time). They do not consider any pre-defined business rules (e.g. the order of the message exchanges). They argue that they can achieve the improvement objective of the process portfolio when each process model within the portfolio has been restructured in order to minimize the processing time.

This approach seems to be structurally correct for both processes in individual context but actually not for the inter-operating business process context including pre-defined rules or constraints consideration since there may be introduced rules/constraints inconsistency within the optimized inter-operating business process. Unfortunately, there is no guarantee to keep the order of the pre-defined rules due to parallel execution on the tasks involved. This problem potentially arises when we do optimization on each process individually for any inter-operating business processes. In addition, there will be a mash problem if we consider to optimize a huge process portfolio with a complex inter-operating relationship among the processes. We will address this in our framework to resolve, i.e. business process portfolio optimization by considering any inter-operating relationship that may be involved in between.

3 Related Work

Ghose and Koliadis [4] present a framework for managing model change by considering each change as a perturbation that leads to inconsistencies and incompleteness to be resolved. Each resolution guides users toward a stable model state for which they have provided declarative characterizations. The authors of this work illustrate the framework with consistency and completeness theories for the graphical BPMN, as well as mapping and completeness constraints for BPMN and Unified Modeling Language (UML)[2] Sequence diagrams. Kindler et al. [5] propose a technique which allows to detect but a few errors of workflow applications which arise from incorrect inter-operation of workflows. They use sequence diagrams to specify the dynamics of the inter-operation of different workflows. Then, they check single workflow for local soundness w.r.t. its sequence specification. If it is locally sound, a composition theorem guarantees global soundness of the complete workflow application.

Different to the other related works, we specifically propose a novel framework for inter-operating business process portfolio optimization. To the best of our understanding, there is no paper which specifically discuss this idea. We focus on not only treating a process as an individual process but also considering the other processes which may have inter-operating relationship with the process being improved. We will focus on determining the most appropriate state (i.e. an optimized state) according to the process objective(s) of a process being changed or improved by keeping the constraint of the process that may exist in accordance with the existing interoperating relationship to the other processes in a process portfolio.

4 Framework for Process Portfolio Optimization

Process optimization mainly correlates with the process improvement, i.e. we will get an *optimized* process from the *non-optimized* one according to the improvement objective(s). Obviously, process improvement relies on the nature of change to business process. Typical process change scenarios influence the greater context of a process in terms of resources involved, goals achieved, compliance rules satisfied, and objectives optimized [6]. Business process models, resource models, constraint (or rules), goals and objectives are five key elements that play a role in business process change management. Evolutionary measures (i.e. minimal change) on a business process improvement would be preferred over more radical reengineering activities [1, 4].

In the following, we will introduce some concepts and definitions required in the process portfolio optimization methodology such as optimization criteria, semantic effect annotation and SPNet (Semantic Process Network).

---

[2] UML Resource Page http://uml.org/

4.1 Optimization Criteria

Optimization criteria assist in defining the objective of the optimization being performed on a particular inter-operating business process model. Optimization criteria allow us to make comparison between process models and to justify a valid optimized process model. Examples of optimization criteria that would be applicable in this framework are processing time, carbon footprint, running cost, etc. These criteria are in the same optimization direction, i.e. minimality such that the smaller value of the selected criteria to achieve, the more optimal the process model. We need to annotate each task with all related information regarding all possibilities of the optimization criteria, e.g. annotating estimated execution time on each task for processing time criterion, and so forth. Consequently, we are not able to perform a machinery optimization based on a particular criterion if the related information on such criterion is not available on each task. A choice of an optimization criterion is problem specific.

However, we also consider the minimal change strategy in order to avoid disruptive improvement on a process. The minimal change strategy and optimization would be in opposite direction. A trade-off must therefore be performed by using a form of *weighted sum* for picking up one process over the others. We would use processing time optimization criterion for the following framework instance.

4.2 Semantic Effect Annotation

Koliadis and Ghose [3] discusses the concept of semantic effect annotation. An effect is the result (i.e. product or outcome) of an activity being executed by some cause or agent. An effect annotation relates a specific result or outcome to an activity on a business process model. Effects can be viewed as both: normative as they state required outcomes; and, descriptive in that they describe the normal, and predicted, subset of all possible outcomes. Effect annotations can be formal (for instance, in First Order Logic-FOL) or informal (e.g. English). In the formal annotation, we assume that effects have been represented in Conjunctive Normal Form (CNF).

An annotated BPMN model is one in which every task and every sub-process has been annotated with descriptions of its immediate effects. For a complete process, we also define a cumulative effect annotation which is obtained from accumulating the immediate effects of all annotated tasks and sub-processes based on all alternative paths to reach a task being observed. Due to space limitation, we are not able to describe in detail and let the readers refer to [3].

4.3 Formal Definition

We borrow a definition from [4] (Definition 1) and give our own definition (Definition 2). We introduce business process proximity to establish a minimality criterion when selecting candidate SPNet revisions [4]. This is required in order to measure the scale of changes compared to the original process model.

Definition 1. [4] Associated with each SPNet is a proximity relation $\leq_{spn}$ such that $spn_i \leq_{spn} spn_j$ denotes that spn_i is closer to spn than spn_j. $\leq_{spn}$, in turn, is defined by a triple $\langle \leq^V_{spn}, \leq^E_{spn}, \leq^{EFF}_{spn} \rangle$ for evaluating node (V), edge (E), and cumulative effect (EFF) proximity respectively. Thus, $spn_i \leq_{spn} spn_j$ iff each of $spn_i \leq^V_{spn} spn_j$, $spn_i \leq^E_{spn} spn_j$ and $spn_i \leq^{EFF}_{spn} spn_j$ holds. We write $spn_i \leq_{spn} spn_j$ iff $spn_i \leq_{spn} spn_j$ and at least one of $spn_i <^V_{spn} spn_j$, $spn_i <^E_{spn} spn_j$ or $spn_i <^{EFF}_{spn} spn_j$ holds.

These relations can be defined in different ways to reflect alternative intuitions. For instance, the following, set inclusion-oriented definition might be of interest: $spn_i \leq^V_{spn} spn_j$ iff $(V_{spn} \Delta V_{spn_i}) \subseteq (V_{spn} \Delta V_{spn_j})$, where $A \; \Delta \; B$ denotes the symmetric difference of sets A and B. An alternative, set cardinality-oriented definition is as follows: $spn_i \leq^V_{spn} spn_j$ iff $|V_{spn} \; \Delta \; V_{spn_i}| \leq |V_{spn} \; \Delta \; V_{spn_j}|$, where $|A|$ denotes the cardinality of set A. Similar alternatives exist for the $\leq^E_{spn}$ relation. Both $\leq^V_{spn}$ and $\leq^E_{spn}$ define the structural proximity of one SPNet to another.

Defining the proximity relation $\leq^{EFF}_{spn}$ is somewhat more complicated, since it explores semantic proximity. One approach is to look at the terminating or leaf nodes in an SPNet (i.e. nodes with no outgoing edges). Each such node might be associated with multiple effect scenarios. For an SPNet spn, let this set be represented by $T_{spn} = \{es_1,...,es_n\}$ where each es_i represents an effect scenario. Let *Diff* $(spn, spn_i) = \{d_1,...,d_m\}$ where d_i is the smallest cardinality element of the set of symmetric differences between $es_i \in T_{spn_i}$ and each $es \in T_{spn}$. In other words, let $S(es_i, T_{spn}) = \{es_i \Delta e | e \in T_{spn}\}$. Then d_i is any (non-deterministically chosen) cardinality-minimal element of $S(es_i, T_{spn})$. Then we write $spn_i \leq^{EFF}_{spn} spn_j$ iff for each $e \in \mathit{Diff}(spn, spn_i)$, there exists an $e' \in \mathit{Diff}(spn, spn_i)$, such that $e \subseteq e'$. This definition exploited set inclusion. An alternative, cardinality-oriented definition is as follows: $spn_i \leq^{EFF}_{spn} spn_j$ iff $\sum |d_i| \leq \sum |d_j|$ for each $d_i \in \mathit{Diff}(spn, spn_i)$ and $d_j \in \mathit{Diff}(spn, spn_j)$.

Definition 2. Given an original process model IP, a process model IP' is defined as preferred (relative to optimization objective O) over another process model IP'' if the following holds:

- IP' and IP'' are compliant (relative to applicable compliance requirements)
- $spn_{IP'} \leq_{spn_{IP}} spn_{IP''}$
- $IP'' <_O IP'$ denotes that IP' performs better under O than IP''.

The optimization objective is represented in the selected optimization criteria. We just provide only single criterion selection to run each optimization objective, not multiple criteria. We operate a minimality form of *weighted sum* to determine the preferred process model IP' over IP'' such that $O'_{IP} < O''_{IP}$, where $O_i = CV_i \cdot w_c + OV_i \cdot w_o$ denoting overall optimization value of process model i, CV_i denoting cardinality of proximity relation of process model i relative to the original process, and OV_i denoting resulting value of selected optimization criteria of process model i. Furthermore, w_c and w_o represent the weight of minimal change and optimization scales respectively such that $w_c + w_o = 1$. The weight value represents the importance of the corresponding parameter over the other, e.g. $w_o > w_c$ denotes optimization is more important than minimal change.

We will do optimization in two contexts, i.e. process optimization in the context of a single process and process portfolio optimization within a process repository. In a single process optimization, we optimize the process by improving the given optimization criteria in order to gain the specified objective(s). If we have successfully got the state that there will not exist another better state after improving the optimization criteria, then we can justify this as a *local optimum*. For example, the best state would be the acceptable fastest time state of the process when we operate a processing time as the optimization criterion. On the other hand, there will exist inter-operating relationship in operating some processes such that one process depends on the other process(s) by sending message(s) in order to accomplish the objectives. If we can optimize each process in a particular inter-operating processes by keeping all pre-defined business rules and achieving the optimization objective(s), we will get a *global optimum* for the whole process.

4.4 Process Portfolio Optimization Methodology

We will consider three main modes of inter-operating business process that are applicable in our framework, i.e. sequential, nested and parallel modes [3]. Figure 2 illustrates these three modes with message exchanges between two processes. We would consider the pre-defined rules in message exchanges order in a nested mode as it is critical.

Inter-operating process portfolio optimization will rely on the following parameters in order to achieve the global optimum result: (1) comply to the pre-defined constraint(s); (2) minimal changes on the model; (3) best achievement to the optimization objective. The first parameter is the critical one to decide whether the processes being evaluated are acceptable to continue to the rest parameter evaluation or not, specifically in a nested mode. Thus, a revised process model will be ignored

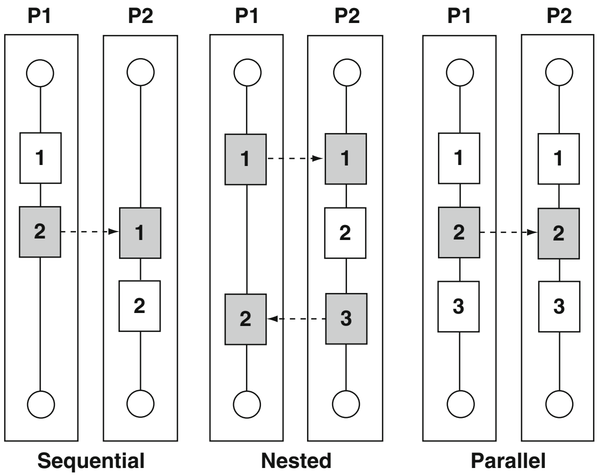


Fig. 2 Three applicable interoperating process modes of processes $P1$ and $P2$

to be an optimized one if it fails to meet the pre-defined constraints. We would use boolean value for justifying this parameter on a particular optimized process. The second parameter is required to justify that the processes being restructured have minimal changes compared to the corresponding original processes. We would use the definition of SPNet proximity relation in order to measure the scale of changes if there were more than one optimum model option such that $spn_i \leq_{spn} spn_j$ denoting spn_i, the optimized alternative i-th, is closer to the original model spn than spn_j, the optimized alternative j-th. The last parameter is the scale to justify whether or not the processes being optimized satisfy to the optimization objective. For example, the smaller number in processing time is preferable when the optimization objective is time processing reduction. The second and third parameters do not represent the order of importance of each parameter since we will recognize both in the same manner by utilizing a form of *weighted sum*.

This framework provides a machinery tool for assisting the analyst to deal with the optimization problem including generating optimized process model variants in such way significantly reduces the degree of his/her involvement by the following two approaches. In order to minimize the number of the generated variants, we introduce τ, i.e. a *threshold* of the overall optimization value O_i. Due to minimality form, the value of τ spans from O_i of the original process (with no change at all) as the maximum value to 0 as the minimum value (although it is not reachable in fact). All variants will be considered as the optimized candidates iff their O_i are less or equal to τ. For both approaches, the analyst involvement is still required to justify whether or not a particular variant should be preferred as the most optimal variant. By using such approaches, we perform optimization in the limited variants available in the variants space.

First approach, this involves exploring i-step changes [7] to provide an optimized process model variant such that a single step corresponds to: (a) determining the *threshold* τ by the analyst; (b) finding one variant satisfies τ by rearranging either all possible existing tasks or all remaining tasks after deletion a task within each individual process by the tool. The task to be deleted must be determined by the analyst before the tool restructures the remaining tasks. i is initialized to one which corresponds to the first variant and incremented in steps of 1 until the preferred variant obtained by the analyst. Once a variant has been found w.r.t. the τ, the rearranging task to produce a variant would be stopped. This approach is similar to hill-climbing local search, where the search landscape is defined by process proximity and optimization objective.

Second approach, this includes a slightly similar way with the previous approach by: (a) determining the *threshold* τ by the analyst; (b) finding all variants satisfies τ by rearranging either all possible existing tasks or all remaining tasks after deletion a task within each individual process by the tool. Once all variants have been found w.r.t. the τ, the rearranging task to produce variants would be stopped.

We propose a methodology for optimizing inter-operating business process portfolio as follows,

1. **Rearranging Tasks within a Process**—Rearrange tasks as much as possible on each individual process within an inter-operating business process model. Rearranging tasks will be depending on the selected criteria to achieve optimization objective, e.g. parallelizing task by using AND gates in optimizing the processing time, or XOR gates in optimizing the carbon footprint, and so forth.
2. **Restructuring Inter-operating Process**—Produce all possible alternatives of inter-operating process by combining the individual optimized processes. Suppose, we have two processes A and B which are in an inter-operating relationship. Let process element A be individually modified into A_1, and process element B be individually modified into B_1 and B_2 in order to optimize. Then, we reconstruct two new inter-operating processes by combining processes A_1 and B_1 as well as A_1 and B_2.
3. **Checking the Pre-defined Compliance**—Verify all possible alternatives against the parameter 1 as Definition 2, i.e. pre-defined rules compliance. The ones that do not meet this criterion must be ignored for the rest evaluation.
4. **Calculating the Minimal Change Measures**—Calculate the measures of difference between the optimized process models and the original one by using the SPNet proximity relation. Each process element within all alternatives of optimized inter-operating business process would be compared with the corresponding process element in the original inter-operating process.
5. **Calculating the Resulting Optimization Value**—Calculate the resulting optimization value based on the selected optimization criteria. For example, we calculate overall processing time of the revised process when we choose processing time optimization criteria.
6. **Obtaining the Solution**—Determine the most optimum of restructured inter-operating process model based on the measures of parameters 2 and 3 as Definition 2.

5 Process Portfolio Optimization Instances

Let us use Fig. 1 to illustrate our framework including *local* and *global optimum* concepts in order to achieve an optimization objective, i.e. reducing the order processing time be less than 23 days from placing the order till receiving goods by the *Customer*. According to the optimization schema, the designer decided to restructuring each process as much as possible on each role *Customer* and *Supplier* as shown in Figs. 3 and 4 respectively. All possible alternative optimized designs have been identified by the designer for both sides. Figure 3 illustrates two options of optimized process on the *Customer* side which is the first option (i.e. Fig. 3(a) to be considered as similar with the original one). Figure 4 illustrates two options of optimized process on the *Supplier* side. Each process has been changed by parallelizing the tasks as much as possible in order to reduce total processing time in each process. By looking at each process individually, the designer argued that Figs. 3 and 4 represent the most possible optimum design to achieve the improvement objective. These are the *local optimum* of each process model.

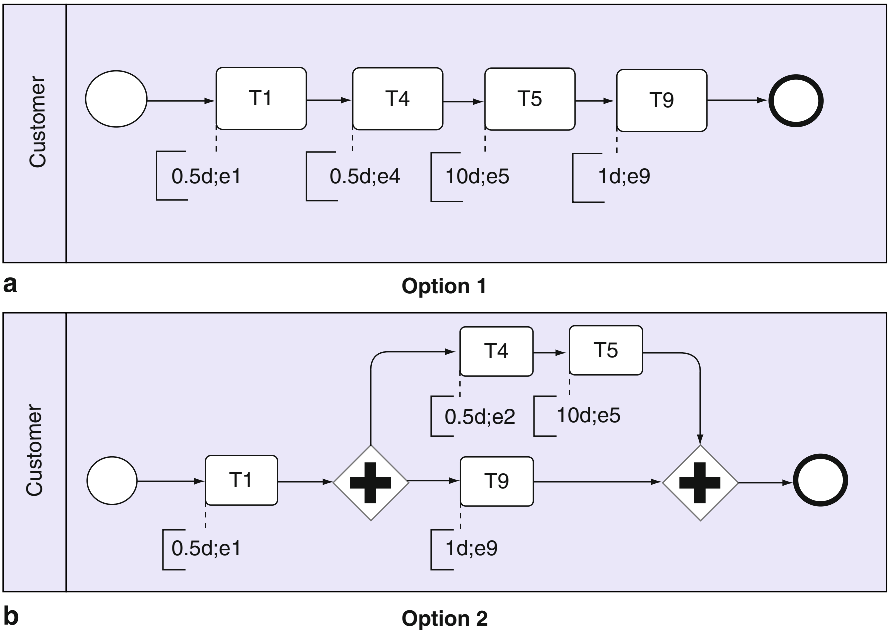


Fig. 3 Two options of individually optimizing the process on *Customer's* side by parallelizing tasks of the original process shown in Fig. 1: either keep the original model or put tasks $T2$, $T5$ in parallel to $T9$

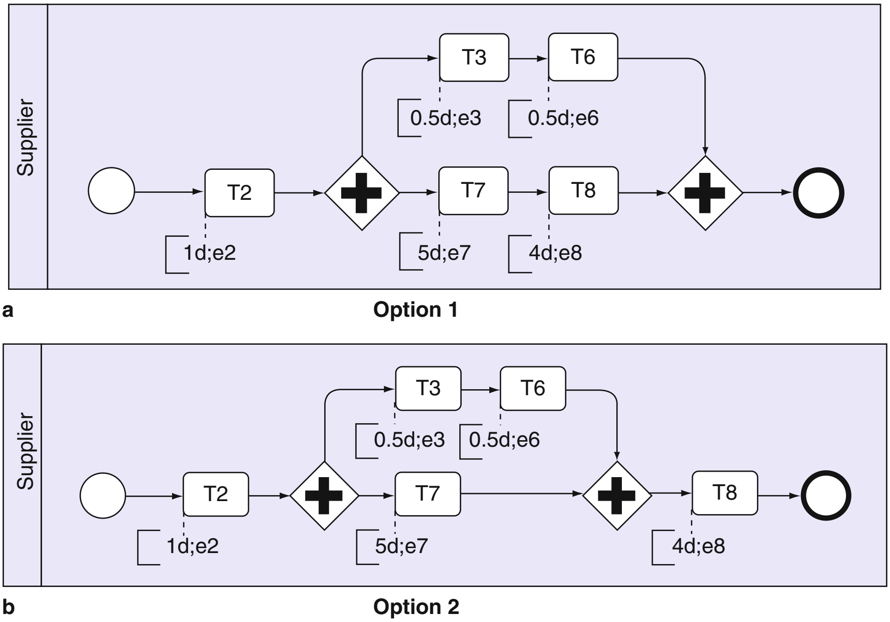


Fig. 4 Two options of individually optimizing the process on *Supplier's* side

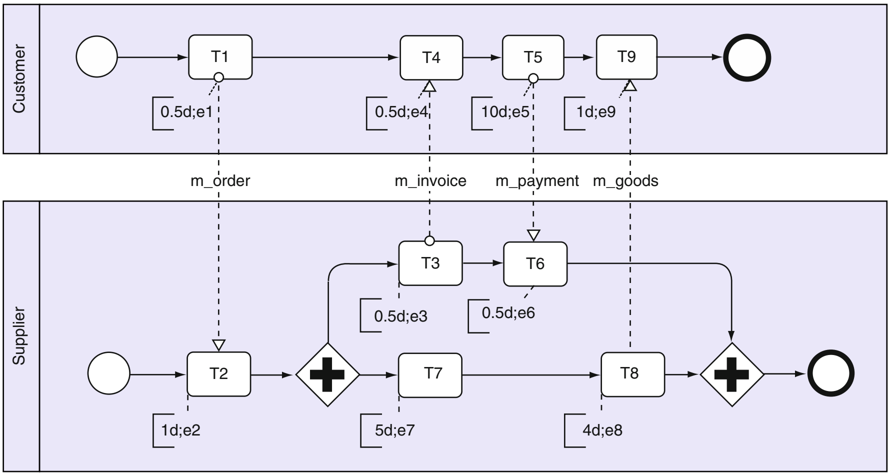


Fig. 5 Alternative 1—combining the first option in Fig. 3 and the first option in Fig. 4. This alternative will be ignored as it does not preserve the order between message *m_goods* and message *m_payment* as defined

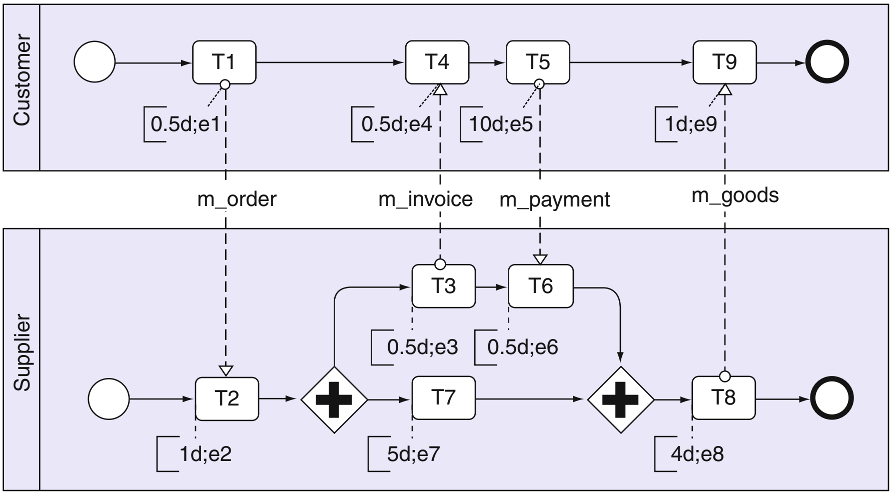


Fig. 6 Alternative 2—combining the first option in Fig. 3 and the second option in Fig. 4. It complies with the pre-defined rule and should be considered as a global optimum candidate

We may consider four alternatives of combining each option in Fig. 3 and each option in Fig. 4 in order to perform a complete order processing function as in Fig. 1. These alternatives are illustrated in Figs. 5, 6, 7 and 8. Let us check the first parameter (i.e. pre-defined constraint compliance such that $m\_order \prec m\_invoice \prec m\_payment \prec m\_goods$). Alternatives 1 and 3, illustrated in Figs. 5 and 7 respectively, would be considered as not comply to the constraint as task *Deliver Goods* will be conducted

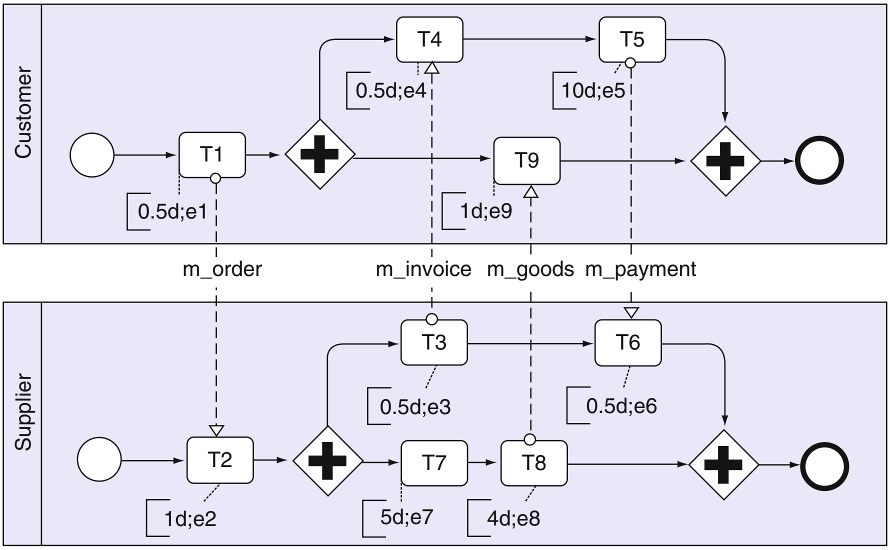


Fig. 7 Alternative 3—combining the second option in Fig. 3 and the first option in Fig. 4. This alternative will be ignored as it does not preserve the order between message *m_goods* and message *m_payment* as defined

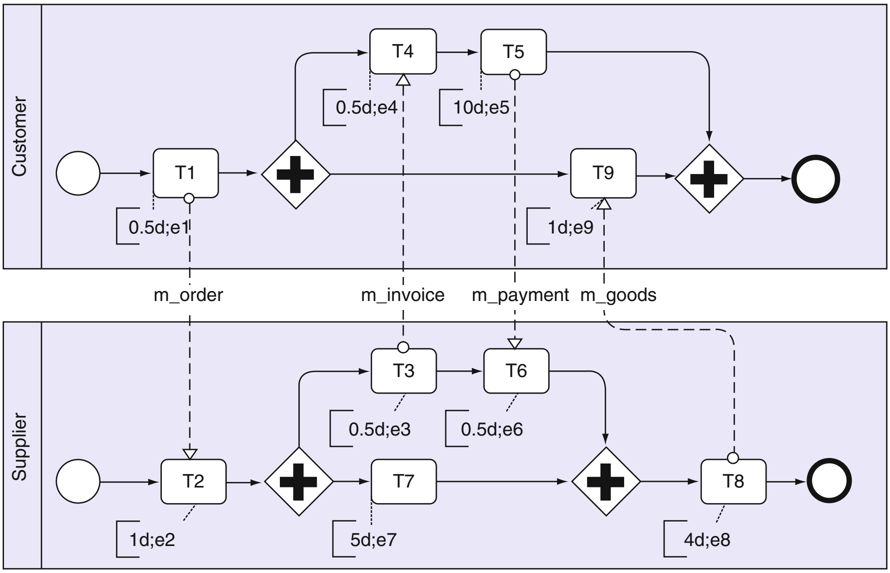


Fig. 8 Alternative 4—combining the second option in Fig. 3 and the second option in Fig. 4. It complies with the pre-defined rule and should be considered as a global optimum candidate

parallel with tasks *Send Invoice* and *Receive Payment* such that message *m_goods* will also be parallel with messages *m_invoice* and *m_payment* respectively. There is no guarantee to keep message *m_goods* will be coming up after message *m_payment*. In such condition, alternatives 1 and 3 would be ignored for the next evaluation such that only alternatives 2 and 4 will be considered for evaluation of the rest parameters.

We apply minimal change inspection on Figs. 6 and 8 compared to Fig. 1 by using SPNet proximity relation. We get 2 nodes of symmetric differences in total between Figs. 6 and 1 due to introducing 2 AND gates on the *Supplier* side process. In addition, there is a significant edge symmetric difference between Figs. 6 and 1 (i.e. 10 edges in total). On the other hand, with the same manner, we have following differences between Figs. 8 and 1, i.e. 4 node differences due to introducing 4 AND gates and 19 edge differences. In regard final accumulative effect of both Figs. 6 and 8 compared to Fig. 1, we would see that Fig. 6 is less difference than Fig. 8.

For the last evaluation, we will check the best achievement to the optimization objective (i.e. reduction in processing time). We calculate 18 days for both Figs. 6 and 8, faster than Fig. 1, from order placed until the goods received. Therefore, no comparison can be made for both alternatives.

Finally, by evaluating these results, the framework will justify that Fig. 6 is the most optimum process model since it complies with the pre-defined constraints, has less changes than Fig. 8 compared to Fig. 1 and achieves 18 days for reducing the overall order processing time.

6 Conclusion and Future Work

In this paper, we have proposed a framework for inter-operating business process portfolio optimization. We have described and illustrated concept of *local* and *global* optimum during process optimization. Optimization on the local context may introduce a serious problem due to inter-operating relationship to another processe(s), i.e. business rules inconsistency. This framework can successfully assist the designer in determining the most optimum process model from several alternatives of local optimized process models based on the defined criteria. Future work includes implementation of this conceptual framework into a semi-automated system for assisting the designer in determining the most optimum revised process in optimizing business process portfolio.

References

1. Weske M (2007) Business process management: concepts, languages, architectures. Springer, New York
2. Rosemann M (2006) Process portfolio management. http://www.bptrends.com/publicationfiles/04-06-ART-ProcessPortfolioManagement-Rosemann1.pdf. Accessed 24 Jun 2009
3. Koliadis G, Ghose A (2007) Verifying semantic business process models in inter-operation. In: Proceedings of IEEE international conference on services computing (SCC 2007), pp 731–738, USA, Jul 2007

4. Ghose A, Koliadis G (2008) Model eco-systems: preliminary work. In: Proceedings of the 5th Asia-Pacific conference on conceptual modelling (APCCM '08), vol 79, pp 19–26, Australia, Jan 2008
5. Kindler E, Martens A, Reisig W (2000) Inter-operability of workflow applications: local criteria for global soundness. Lecture Notes in Computer Science, vol. 1806, Business Process Management, pp 129–166
6. Koliadis G, Ghose A (2009) A conceptual framework for business process redesign. Lect Notes Bus Inf Process 29:14–26
7. Ghose AK, Koliadis G (2008) PCTk: a toolkit for managing business process compliance. In: Proceedings of the 2008 international workshop on Juris-Informatics (JURISIN-2008), Japan, Jun 2008

Reconciling Usability and Security: Interaction Design Guidance and Practices for On-Line User Authentication

Michael Lang

Abstract Usability and security are often portrayed as though they are competing priorities in information systems development. Given that both are essential to the design of an effective system, it is important that these two prerogatives should be reconciled. In recent years, there is growing concern with the rising incidence of on-line impersonation, theft and other types of fraud. It is therefore important that an information system must have a secure and rigorous way of authenticating a user's identity. This paper reviews the sources of literature on interactive design guidance for on-line user authentication, and then compares the actual practices of a purposefully selected sample of twelve Websites against the recommendations from the literature. Alarmingly, the findings of this study are that many Websites have user authentication processes which contain basic design flaws that are potentially open to exploitation by Internet criminals.

1 Introduction

Government statistics reveal that the US Department of Justice received over 330,000 complaints of Internet crime in 2009, up 22% on the previous year, causing estimated losses of US$560m as opposed to US$265m in 2008. Of the complaints which were subsequently referred to law enforcement agencies, most related to e-commerce fraud, with identity theft (14%) and credit card misuse (10%) being notably to the fore [10]. Comparable figures released by the Data Protection Commissioner in Ireland indicate a similar trend (see Fig. 1), leading him to comment that "*respect for privacy is part of the network of trust that our society relies upon … a collapse of public trust in data-dependent services organisations would be hugely damaging*" [4]. Data security has long been recognised as an important issue in systems analysis and design [2], but given the modern situation where e-commerce and e-government

M. Lang (✉)
J.E. Cairnes School of Business and Economics, National University of Ireland, Galway, Ireland
e-mail: michael.lang@nuigalway.ie

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_32,

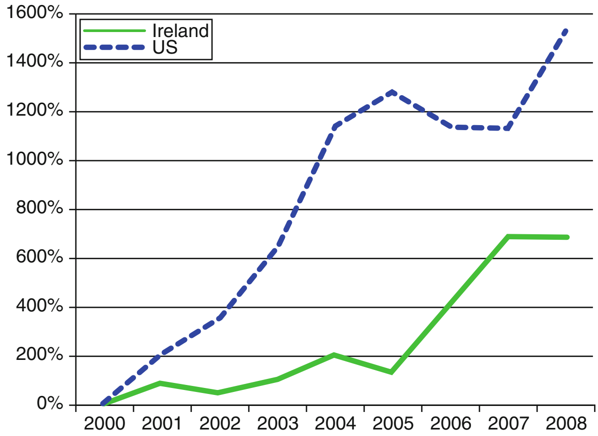


Fig. 1 Growth in incidence of recorded Internet crime complaints since 2000. (Sources: [4, 10])

systems which store and process private personal data are plugged into an expansive global communications network, security has now become a critical factor in Web-based systems development. In order to combat fraud, it is necessary to be able to authenticate the identity of a user with a high degree of confidence. The essential challenge is how to design a system that is both usable and secure while also respecting users' privacy [17].

The joint prerogatives of information systems security and usability can sometimes appear to be in conflict with one another [3, 25, 28], with Zurko and Simon [30] going as far as to say that "secure systems have a particularly rich tradition of indifference to the user". The more security controls that a legitimate user is required to pass through in order to authenticate himself, the more likely that user is to attempt to circumvent those barriers in order to minimise the effort required to access the system, perhaps even subverting the integrity, security and effectiveness of the system [15, 27, 28]. For example, free Web sites such as www.fakenamegenerator.com and www.bugmenot.com are specifically intended to bypass the registration process of e-commerce systems by providing false details, the reason being that sites which require compulsory registration are perceived by many users as a nuisance and an invasion of privacy. There is thus a need to strike a balance the vital considerations of security, usability, privacy, and data integrity, and with this in mind Norman [17] has called for "*a set of standardized scripts, templates, and system tools that allows [the community] to implement best practices in ways that are both effective and efficient, standardizing interactions across systems in order to simplify the lives of users*". The motivation for this paper is to go some way towards answering that call. Information systems security is multi-layered (e.g. the OSI security architecture), but because the focus of this paper is on user interaction, we shall concern ourselves only with the design of the front-end of the system, concentrating on the basic tasks of registration and logging in, both of which are central to user authentication and fraud prevention.

The structure of this paper is as follows: Sect. 2 reviews the literature on interaction design guidance for on-line user authentication; Sect. 3 outlines the research approach; and Sect. 4 discusses the findings of an analysis of the actual practices used by a selection of Web sites, comparing those practices against the recommendations in the literature. Only one previous study has conducted a similar analysis [7], with most of the published work in the area of user interaction design for on-line authentication being of a theoretical or prescriptive nature. This paper therefore contributes towards building a better understanding of the fit between textbook theory and industry practice.

2 Literature Review and Commentary

Preece et al. [18] define "interaction design" as the process of "*designing interactive products to support people in their everyday and working lives*". It is closely associated with the notions of "user experience design" and "information architecture", particularly within the context of Web-based systems development. Underpinning all of these is a central concern on usability, the essential aspects of which are: effectiveness, efficiency, utility, learnability, memorability, and safety. The practical discipline of interaction design is informed by a body of design guidance based on experience and theoretical knowledge accumulated within its own community and also inherited from cognate fields (e.g. industrial design). This body of design guidance encompasses knowledge in various formats, including design principles, design patterns, and use case scenarios.

Design principles are "generalizable abstractions intended to orient designers towards thinking about different aspects of their design" [18]. These general principles, such as "visibility" and "affordance", lead to what may be called design "heuristics" in practice, meaning the way a particular problem can be resolved (e.g. [28]). The notion of a "design pattern", first originated by Alexander [1] in the field of architectural design, refers to a way of formulating the description of a commonly recurring design problem as well as the core of the recommended solution to that problem. A design pattern typically comprises the following sections, though it need not include all of these: name, aliases, classification, motivation, problem statement, forces/rationale/intent, solution/sequence of events, participating patterns/classes, examples/known uses, consequences, and related patterns.

Design patterns have been around for some time within the field of software engineering (e.g. the PLoP conference series) and are now also popular within the field of human-computer interaction (see for example UI-patterns.com). For Web-based applications development, most of the patterns in the literature are primarily concerned with hypermedia navigation, various aspects of interface design, software engineering, or programming (e.g. [6, 9, 13, 15]), but there also exists a number of valuable contributions which describe interaction design patterns for on-line user authentication [16, 22, 23, 29].

Similar in a number of regards to a design pattern is the notion of a "use case". This is a requirements specification technique commonly used in user-centred design approaches and is a normal feature of UML-driven development approaches. A use case scenario is a description of a particular task that an actor is required to perform, generally written up in a standard format which includes: use case name, participating actors, triggers, pre-conditions (if any), flow of events (including exception handling), post-conditions (if any), and relationships with other use cases (e.g. "includes", "extends"). For standard functionality that is similar across most systems, use case scenarios can provide a reusable source of design guidance. Of particular interest is the idea of "misuse" or "abuse" cases, where the malicious behaviour of a fraudster is anticipated when designing the system [14, 20].

The sections which follow summarise the various sources of interaction design guidance found in the literature on the two basic tasks associated with on-line user authentication: user registration, and logging in.

2.1 User Registration Process

In order to permit access to the secure areas of an information system, it is necessary to be able to authenticate a user's identity so the user must be registered. Out of respect for users' trust and privacy, Egger [5] recommends that the imperative to register should be delayed as long as possible e.g. by firstly letting a user browse the site and add items to a shopping cart as a guest user. This is referred to as the *Lazy Registration* pattern (see http://ui-patterns.com./pattern/LazyRegistration). When the user is eventually asked to create a profile, a full overview and justification for all required data attributes should be clearly provided and the registration process should collect the minimum amount of information needed [22, 23]. To prevent registration on a Web site by a fraudulent automated agent as opposed to a human, the use of a CAPTCHA is recommended. A further precaution is not to activate a user's account until it has been verified, which is normally done by sending a confirmation request to the email address provided during registration [23].

As regards user authentication, the two essential attributes of the user's profile created upon registration are the password and the UserID, so we shall concentrate on design considerations impacting these two attributes.

In a previous study of Website password practices, Furnell [7] asserts that it is reasonable to expect users to be provided with guidance on how to choose a secure password, but most of the sites that he assessed were remiss in this regard, thus "leaving users to select passwords in a potentially *ad hoc* manner". There is general consensus within the literature on the basic principles of what constitutes a "strong" password, as well as on simple practices to protect passwords from being discovered easily:

- It is recommended that passwords should be at least 8 characters long [7, 24, 27], but numerous articles mention the trade-off between password complexity/length and memorability [11, 26, 27]. Users are more likely to write down passwords that they cannot remember, thus potentially compromising security [31]. As yet, no research has been conducted to investigate if the Password Manager feature of

Web browsers has had any effect on the tendency to choose short passwords or to write longer passwords down.

- The password should ideally use a mixture of upper and lower case letters, numbers, and symbols [24, 26, 27]. In theory, extending the password length has a substantially greater impact on strength than just increasing the number of available characters [11], but this only necessarily holds if the password is a random sequence of the available characters.
- The use of passphrases (e.g. “EatSleepDrinkFootball-24-7!”) as opposed to passwords has been found to increase password strength while still being memorable [11]. Alternatively, a passphrase can be used as a mnemonic to recall an alphanumeric password e.g. “I have to catch a plane to London at ten o’clock“Ih2cap2L@10oc” [7, 24, 26].
- Avoid simple patterns, such as a capitalised letter at the beginning or placing digits at the end [24].
- Reject passwords that can be found in precompiled dictionaries [7, 11]. For 4-digit PINs, avoid obvious combinations such as “1234”, and for 6-digit PINs avoid the temptation to base the PIN on an obvious DDMMYY date (e.g. the user’s birthday).
- A particularly bad choice for a password is “password”, but remarkably Furnell [7] has found that quite a number of major e-commerce Websites permit this.
- The risk of a “user information attack” can be reduced by avoiding passwords based on discoverable facts such as a person’s date of birth, child’s name, football team etc. [7, 26, 27].
- Users should be forced to change their passwords regularly, at least once every 3 months [7, 26, 27]. This also helps with password memorability [31]. The corollary is that organisations should deactivate dormant accounts.
- Do not permit the reuse of previous passwords because these may have been compromised [7].
- As a password reminder “hint” for the purposes of reset/recovery, allow a user to define his own secret question [7], as opposed to using questions based on discoverable facts (e.g. place of birth) or facts that might be easily guessed (e.g. most people’s favourite film is near the top of the Internet Movie Database chart at www.imdb.com/chart/top). This approach is more secure than sending forgotten passwords by email, which is certainly not recommended for accounts containing sensitive information [22].

In contrast to the wealth of guidance on what constitutes a strong password, there is comparatively little advice on how to choose an appropriate UserID (assuming it is not assigned by the system). Vora [23] suggests that when registering for Websites, a user’s email address is often a good choice for a UserID on the basis of uniqueness and memorability. However, there is a considerable security risk attached to the common practice of using an email address as a UserID. Users habitually use the same password for multiple accounts, and rarely change their passwords [12, 17, 31]. This problem is therefore compounded if users also have the same UserID for multiple Websites [7, 11]. For example, a spoof Website offering the prospect of last minute bargains on unsold concert tickets could deceive a user into registering, but could then try to use the same UserID and password to log in to other sites, including the user's email account. A better resolution to the problem of a user having multiple Websites with the same UserID and password is to permit users to register using a "unified" service such as Windows CardSpace, SAML or OpenID [23]. Of course, if a user can choose his own UserID, there is no barrier in theory to the use of the same principles as previously outlined so as to create a "strong" UserID.

2.2 System Login Process

In their seminal paper, Yoder and Barcalow [29] describe a design pattern named *Single Access Point* (alias *Login Window*), the basic premise of which is that there must only be one way into a system; otherwise put, everybody must enter via the front door as opposed to side and back doors. The standard solution is to create a single login screen through which all users must pass in order to gain authorised access to the restricted areas of the system. If a user attempts to access the system through a different access point, such as attempting to go directly to the URL of a restricted area, the user should be redirected through the normal login process—this simplifies control flow and encourages modular design by creating a reusable login "use case". A closely related pattern is named *Check Point* (alias *Authentication and Authorization*), which verifies the credentials of a user seeking to pass through the login process. The *Check Point* pattern recognises that users will occasionally make mistakes, such as entering an incorrect password, and that different actions need to be taken depending on the severity and number of mistakes. The recommended solution is to "create an object that encapsulates the algorithm for the company's security policy" [29] by considering every branch in the authentication logic, such as how to handle password failure or password expiration. Once a user's login is authenticated, the *Session* and *State* patterns then come into effect.

These general principles are by now well established, and in Web application design a user's login state is typically maintained through the use of cookies and HTTP sessions. A factor that must be considered is whether to permit a user to remain permanently logged in, or whether to log a user out after a certain period of inactivity [23]. Although it may be convenient for a user to remain continuously logged in to a system, this could raise a potential security problem if somebody else were to gain access to that user's workstation. Yee [28] emphasises the interaction design principle of "path of least resistance", meaning that a system should be designed so that a user cannot inadvertently compromise the default secure state of a system (e.g. by not logging out).

Although the implementation of the *Check Point* pattern will vary from one organisation to the next, many of the issues that need to be considered are common. For example, how many times should a user be permitted to enter a password incorrectly before the user's account is deactivated? Password cracker algorithms often use "brute force" attacks based on dictionary words or other known common combinations e.g. sequences of adjacent keyboard characters, or substitutions of digits for letters such as "4" for "A" or "0" for "O". Keith et al. [11], working from the previous findings of Zviran and Haga [31], estimate that an unrestricted brute force attack could crack a typical password in just two seconds. Therefore, to prevent such attacks it is important to stipulate what could be regarded as a reasonable number of incorrect login attempts [19]. Vu [24] recommends a "six strikes and you're out policy" for failed logins as opposed to the common practice of permitting three strikes, his rationale being that users often have many passwords to remember and therefore can mix them up.

Of course, one cause of repeated failed login attempts may be that a user has forgotten the login details, or there may be a problem with case sensitivity. In most cases, the latter issue can be handled simply by reminding the user that the password is case sensitive and by having a feature which warns if CAPS LOCK is on. As regards forgotten login details, it is important that the system has a process for handling lost passwords [22] as an extension of the normal login use case. Most systems have a "Forgot password" link which activates a reset or recovery mechanism, perhaps simply emailing the password (an insecure approach), or by emailing a once-off URL reset link, or else by asking the user a series of secret questions. Depending on an organisation's security policy, some systems may prevent users from changing their password unless personal contact is made with a human representative. On the basis of the author's personal experience, it is potentially risky not to have a mechanism to permit a user reset his password or at least temporarily disable his account without the need for the intervention of a system administrator: for example, if an organisation uses single sign-on to all its systems, an out-of-normal-hours compromise of any of those systems could place them all at risk if the password were not quickly changed.

Thus far, authentication has been discussed from the perspective of an organisation seeking to verify the identity of a user trying to access that organisation's resources. However, authentication is a "two-way street" and the user is also entitled and should be encouraged to verify that the party with whom he is interacting is not an impostor. Yee [28] refers to this as the "principle of the trusted path". For example, a user could receive an email or SMS message asking him to go to a bogus Web site or to call a telephone number, whereupon the user is asked some "standard security questions"; this is referred to as "phishing" and is one of a number of so-called "social engineering" techniques that confidence tricksters engage to lure users into disclosing personal details. Most password breaches arise this way, rather than through crude brute force attacks [17].

Unfortunately, the dilemma with Web security is that many of the issues which cause the most problems are not directly controllable. User interaction designers are limited in what they can do to prevent on-line fraud, firstly because the user's Web browser has inherent vulnerabilities (e.g. cookies, stored passwords, AutoFill options) that can potentially be exploited by malware, and secondly phishing/spoofing hoaxes can be so elaborate as to almost exactly emulate the look-and-feel of an actual Website (e.g. by registering a similar domain name, substituting visually-close characters such as capital "I" or the digit "1" for the letter "l" [25]). Thus, as Internet criminals engage in ever more devious means of entrapping users, the challenge for user interaction designers is how to maintain public trust by outwitting the fraudsters [17, 19].

Egger [5] makes a number of recommendations as to how security and privacy issues can be handled in order to build and sustain a sense of trust. Amongst these are: prominent links to applicable policies (e.g. data protection, privacy/security, consumer rights), describe the technologies being used to assure security (e.g. SSL), and use trusted third parties (e.g. payment options, security certificates). Vora [23] and van Duyne et al. [22] also emphasise the importance of having a clearly visible link to the organisation's privacy policy. Additionally, it is useful to provide advice to users about how to prevent and detect common types of fraudulent Internet "scams".

"Spyware" applications, "sniffers" and "Trojan horses" can potentially negate the use of Secure HTTP/SSL if they log keystrokes or grab passwords within the Web browser client *before* they are encrypted [19]. To overcome this, some financial institutions use hardware tokens to authenticate each transaction, but this places a burden on the user because simple tasks can become cumbersome [25]. Another mechanism which is being increasingly used to confound spyware is a CAPTCHA, an image-file of a randomly-generated character string which is readable by humans but not by computers [8].

Passwords can also be stolen by "shoulder surfing", where a fraudster observes a user entering his password either by loitering in close proximity or else through the use of an inconspicuous recording device. To mitigate this risk, passwords are usually masked on entry [23]. However, because the user cannot see what he has entered, two problems can arise with masks: firstly, the user may enter an incorrect password (especially if it is case sensitive); secondly, password-grabbing spoof Web sites can exploit the mask by luring a user into blindly disclosing all the digits of a PIN (e.g. "Please enter 1st, 2nd, and 5th digits" → "Sorry, PIN failed. Please try again" → "Please enter 3rd, 4th, and 6th digits" → "System temporarily unavailable, please try again" → re-direct to actual login screen). One potential improvement here would be if a user were given the option each time of whether or not to mask the characters entered in a password input box. Conversely, if there is a risk of a login process being observed, it would make sense to give the user the option of also concealing the UserID field with a mask.

3 Research Approach

Having reviewed the literature on interaction design guidance for on-line user authentication, the empirical phase of this study involved a comparison of actual practices used by Websites against the practices recommended by the literature. A purposefully selected sample of 12 Websites was chosen so as to cover a variety of Website types commonly targeted by Internet criminals, and also so as to include organisations from a number of different countries (Ireland, UK and USA/International). Because a proper analysis of Website security practices requires a longitudinal approach (e.g. to assess if periodic password changes are required), these sites were also chosen on the basis that the author is a habitual user with experience of having used all of them over an extended period of time:

- **Social Networking/Personal Communications (4):** Twitter, Gmail, Facebook, and LinkedIn.
- **On-line Banking/Insurance Services (3):** Bank of Ireland 365online (www.365online.com), Quinn Healthcare (www.quinnhealthcare.com), and Rabobank Online (www.rabodirect.ie).
- **Electronic Commerce (3):** eBay, PayPal, and Amazon UK (www.amazon.co.uk).
- **Electronic Government (2):** PAYE anytime (www.ros.ie), where Irish citizens can manage their income tax returns, and Motor Tax Online (www.motortax.ie).

The sites were assessed by creating a new "dummy" user registration (using an identity generated by www.fakenamegenerator.com), and where this was not possible an existing account to which the author had prior access was used. After initial registration, the sites were evaluated by following the available processes for logging in, retrieving lost/forgotten login details, and changing the user profile. Basic deliberate errors were made to test the security features, such as attempting to log in repeatedly with an incorrect combination until access is blocked, going directly to the URL of a restricted access area which requires prior login authentication, choosing a weak password, and leaving a user logged in but inactive for a period of time. The sites were also assessed for compliance with the norms of good practice as outlined in the previous section, such as clear links to appropriate privacy/security policies and periodic forced password changes. This research was conducted in April 2010. A number of the Web sites studied have since modified their security features.

4 Discussion of Findings

4.1 Assessment of Design of Password Mechanisms

The legend used in Table 1 for the types of characters allowable in passwords is as follows: 'A' signifies an upper case letter, 'a' signifies a lower case letter, '9' signifies a numeric character, '%' signifies a symbol character (i.e. neither a number nor a letter), and '+' specifies that the particular character type is required. The allowable types of characters are enclosed within square brackets []. If there is a stipulation that the password must contain at least 'n' different types of characters this is indicated by '(n)' after the square brackets. For example, if a password is of type [A,a,9+,%](2), this means it must contain at least one digit and also either an upper or lower case letter or symbol. If no restrictions are placed on the characters to be included in a password, this is indicated by [*]. Most of the Websites evaluated required that passwords be a minimum of 6 characters long, comprising a mixture of letters, digits and symbols. Three sites, Bank of Ireland, PAYE anytime, and Motor Tax Online, are accessed using a combination of facts and PIN, while the interaction with Rabobank Online is mediated using a physical hardware token. To test for the strengths of passwords permitted by each of the Websites, some typical choices were tried, including a very weak password ("abcd1234"), a 16-character dictionary word ("UnpredictabilitY"), a passphrase from the children's tale of Ali Baba ("OpenSesame"), a keyboard sequence in reverse ("Poiuytrewq"), a repeating 16-digit numeric sequence ("0123456789012345"), a variation of the initials and date of birth of the dummy user ("CEB300457"), the name of a football club and squad number of its star player ("Liverpool9"), and an abbreviation of an address ("1600PennAve"). The results are displayed in Table 2, and it is interesting to compare these findings against the password selection guidance provided on the sites. Although most of the sites warn against choosing a password based on a dictionary word, only one of them actually forbids this (QuinnHealthcare), this being because that is the only Website to insist that the password must contain a non-alphabetical character. Only QuinnHealthcare and eBay have adequate checks to ensure that a mixture of character types must be used, and also that neither a UserID nor a person's name can be used as his password. All of the Websites were case sensitive.

Gmail did not permit the UserID to be used as a password, but did allow the selection of a password based on the user's name with initial capitals; its password strength meter even indicated that this was 'Strong'. The passwords listed in Table 2 flagrantly ignore all of the password selection tips in Gmail's checklist of "Things to Avoid", but nevertheless Gmail only rejected "abcd1234", with "OpenSesame" getting a "Good" rating and all of the others rated as "Strong". In contrast, the Facebook meter disallowed not just the obviously weak "abc1234", but also rejected "OpenSesame" and "Poiuytrewq". Both "UnpredictabilitY" and "0123456789012345" were given a meter reading of "Medium" but were still permitted, and the others were all deemed "Strong". LinkedIn, Amazon and PayPal do not have password strength meters and they accepted all passwords, as did Twitter, flying in the face of their own password selection advice:

- PayPal: "*We recommend that your password is not a word you can find in the dictionary, includes both capital and lower case letters, and contains at least one special character (1–9, !, *, _, etc.).*" In actual fact, PayPal accepts any combination of eight or more characters of any type.
- LinkedIn: "*A good password should contain a mix of capital and lower-case letters, numbers and symbols*".
- Twitter: "*Be tricky! Your password should be at least six characters and not a dictionary word or common name. Change your password on occasion.*"
- Amazon: "*Use at least eight characters, a combination of numbers and letters is best; Do not use the same password you have used with us previously; Do not use dictionary words, your name, e-mail address, or other personal information that can be easily obtained; Do not use the same password for multiple online accounts.*" Peculiarly, Amazon only provides advice when changing a password, not when creating an account. Only the first of these five recommendations is enforced by Amazon.
- eBay: "*Passwords must have a mix of letters, numbers and symbols... do not use a dictionary word.*" However, eBay is willing to accept a dictionary word, as long as it is of mixed case.

Table 1 Summary of analysis of design practices used by Web sites in practice

✓ Guideline implemented ✗Guideline implemented "N/A" = not applicable	Electronic government		Social networking/Personal communications				On-line banking/Insurance services			Electronic commerce		
	PAYE anytime	Motor Tax Online	Google Gmail	Twitter	Facebook	LinkedIn	Bank of Ireland	Rabobank Online	Quinn Healthcare	Amazon (UK)	PayPal	eBay
"Lazy Registration" facilitated?	N/A (demo)	N/A	N/A	N/A	N/A	N/A	N/A (demo)	N/A (demo)	N/A	✓	N/A	✓
Verification of registration? (email or offline)	✓ (off-line)	N/A	✗	✓	✓	✓	✓ (off-line)	✓ (off-line)	✗	✗	✗	✓
Use of CAPTCHA to register?	✗	N/A	✓	✓	✓	✗	N/A	N/A	✗	✗	✗	✓
Use of CAPTCHA to change password?	✗	N/A	✗	✗	✗	✗	✗	N/A	✗	✓	✗	✗
Password type	PPSN+ fact+ PIN	Vehicle Num. +PIN	[*]	[*]	[*]	[*]	UserID+ fact+PIN	"DigiPass" device	[A,a,9+] (2)	[*]	[*]	[A,a,9, %](2)
Password length (min, max)	6 digit PIN	6 digit PIN	8,100	6,-	6,-	6,16	6 digit PIN	N/A	6,-	8,20	8,40	6,20
Password/PIN strength check?	✗	N/A	✓	✓	✓	✗	✗	N/A	✗	✗	✗	✓
Password/PIN strength meter?	✗	N/A	✓	✓	✓	✗	✗	N/A	✗	✗	✗	✗
Guidance on password selection?	✗	N/A	✓	✓	✓	✓	✗	N/A	✗	✗	✓	✓
Forbids password based on word?	N/A	N/A	✗	✗	✗	✗	N/A	N/A	✓	✗	✗	✗
Forbids password same as UserID or person's name?	N/A	N/A	✗	✗	✗	✗	N/A	N/A	✓	✗	✗	✓
Password case sensitive?	N/A	N/A	✓	✓	✓	✓	N/A	N/A	✓	✓	✓	✓
Input masking on password?	PIN only	✓	✓	✓	✓	✓	PIN only	N/A	✓	✓	✓	✓

Table 1 (continued)

✓ Guideline implemented ✗ Guideline implemented "N/A" = not applicable	Electronic government		Social networking/Personal communications				On-line banking/Insurance services			Electronic commerce		
	PAYE anytime	Motor Tax Online	Google Gmail	Twitter	Facebook	LinkedIn	Bank of Ireland	Rabobank Online	Quinn Healthcare	Amazon (UK)	PayPal	eBay
Forbids re-use of previous password?	✗	N/A	✗	✗	✗	✗	✗	N/A	✗	✗	✗	✓
Forbids use of email address as UserID?	N/A	N/A	✗	✗	✗	✗	N/A	N/A	N/A	✗	✗	✓
"Single Access point"/"Check Point"?	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
User automatically logged out after a few minutes of inactivity?	✓	Not ascer-tained	✗	✗	✗	✗	✓	✓	✗	✗	✓	✓
No. of incorrect logins permitted	Not tested	Numerous (25+)	10	3	3	10	3	Not tested	Not tested	Numerous (25+)	10	3
Forces periodic password change?	✗	N/A	✗	✗	✗	✗	✗	N/A	✗	✗	✗	✗
Facility to recover lost password?	✓ (off-line)	N/A	✓	✓	✓	✓	✗ (off-line)	✗ (off-line)	✓	✓	✓	✓
Avoids recovery hints based on easily discoverable facts?	N/A	N/A	✓	N/A	N/A	N/A	N/A	N/A	✗	N/A	✗	✗
Permits user-defined recovery hint?	N/A	N/A	✓	✗	✗	✗	✗	N/A	✗	N/A	✗	✗
Links to privacy/security policies?	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Description of secure technologies?	✓	✓	✗	✗	✗	✗	✓	✓	✓	✗	✓	✓
Obvious link to fraud advice?	✓	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✗

Table 2 Results of tests of the admissability of various password types

✗ Disallowed password ✓ Permitted password	Google Gmail	Twitter	Facebook	LinkedIn	Quinn Healthcare	Amazon (UK)	PayPal	eBay
abcd1234	✗	✓	✗	✓	✓	✓	✓	✗
0123456789012345	✓	✓	✓	✓	✗	✓	✓	✗
1600PennAve	✓	✓	✓	✓	✓	✓	✓	✓
Liverpool9	✓	✓	✓	✓	✓	✓	✓	✓
CEB300457	✓	✓	✓	✓	✓	✓	✓	✓
Poiuytrewq	✓	✓	✗	✓	✗	✓	✓	✓
OpenSesame	✓	✓	✗	✓	✗	✓	✓	✓
UnpredictabilitY	✓	✓	✓	✓	✗	✓	✓	✓

Bank of Ireland 365online uses a "multiple input" approach in an effort to strengthen its login combination. A user is required to enter three pieces of information to access his account: a unique six-digit user ID, a personal fact (usually date of birth or the last four digits of a telephone number), and three random digits from a user-defined six-digit PIN. Of these three pieces, the second and third are vulnerable. A date of birth and telephone number can be easily compromised e.g. through a Facebook profile, phishing attacks of CVs by spoof recruitment agencies, or simply through the loss of a handbag. Six-digit PINs are vulnerable because people tend to base them on memorable DDMMYY dates. For example, if a hacker guessed that the six-digit PIN was based on a date between 1980 and 2009, and was asked to enter the 1st/3rd/5th digits, then the respective possibilities are [0,1,2,3]/[0,1]/[8,9,0], giving a maximum of 24 permutations. Given that the likelihood of the first digit being 3 is low (4.9% i.e. only 18 possibilities out of 365 days), then the number of permutations is further reduced.

In contrast to the Bank of Ireland approach, Rabobank uses a physical "Digipass" device which generates a token to authenticate the various steps of a transaction. However, simple processes such as transferring funds between accounts are substantially more cumbersome in the Rabobank system than with Bank of Ireland. It is well acknowledged within the substantial body of literature on users' attitudes towards technology adoption and usage that perceived effort and perceived utility tend to be closely correlated, so users may be deterred by the "Digipass" device if they feel that the marginal gain in security that it offers is offset by the additional effort required to operate it.

The login combination for the *PAYE anytime* system is very similar to Bank of Ireland, except that in place of a bank customer UserID, it uses an individual's Personal Public Service Number (PPSN). This is a unique identity number given to each citizen in Ireland, which can be found on documents such as payslips, pharmacy prescriptions and wallet cards. Thus *PAYE anytime* is more vulnerable to an attack than the Bank of Ireland system because the three pieces of required information are either relatively easy to discover (i.e. PPSN, telephone number) or guessable (i.e. a weak PIN based on a significant date). This is all the more serious because *PAYE anytime* grants a user access to the income tax returns of himself and his partner for the previous 6 years, and stores details of names, dates of birth, address, telephone numbers, email, employment details, and bank account details.

Of all the sites evaluated, the one which contained the most sensitive information was QuinnHealthcare, an insurance provider. Once logged in, a person has access to extensive private data about a policy holder and his dependents, including names, genders, dates of birth, address, telephone number, email, PPSNs, and bank account details. Alarmingly, this was one of the most insecure Websites visited. The interaction design of this system is very unusual in so far as the very same details are required to register as to retrieve a forgotten password. In both cases, what is required is the user's membership number and date of birth, details that can easily be accessed by a criminal by stealing a user's wallet or robbing a medical centre. Another poor aspect of this site's design is that there appears to be no upper limit on the number of characters that can be entered as a password: a repeating sequence of 500 characters generated a “Password saved” message, but the subsequent login failed, most likely because the password was truncated when stored in the underlying database field.

4.2 Review of Security Checks for Login and Logout

The policies regarding login timeouts varied across the twelve Websites. Bank of Ireland, Rabobank, PayPal, eBay and *PAYE anytime* all force a user to log back in if he has been inactive for a few minutes. This is understandable given that these five Websites are all in the financial services or e-commerce sector. On the other hand, Gmail, Facebook and a number of other sites provide the user with an option to remain permanently logged in. The rationale for allowing habitual users of a system to remain continually logged in is to make the user experience hassle-free, but the cost-benefit of requiring a user to re-authenticate himself at least every few hours must be evaluated against the threat of the user’s account being accessed illegitimately, as could for example happen if it were left temporarily unattended in an open workspace. eBay has a happy middle-ground where a user can elect to remain logged in for 1 day, but the logged in state does not persist beyond that period.

Another important user interaction design consideration is how to handle multiple consecutive failures to enter the correct login combination. Bank of Ireland and eBay both handle this by blocking access to an account after three failed login attempts. Facebook also permits a user to have three clear strikes, but then the user must decipher a CAPTCHA to request that a password reset code be sent to his email address. Twitter follows much the same idea as Facebook.

A number of other Websites allow a user have ten incorrect login attempts. Thereafter, PayPal blocks access and requests a user to make a telephone call to Customer Service. With both Gmail and LinkedIn, every subsequent attempt beyond the tenth consecutive failure requires a user to decipher a CAPTCHA in addition to entering the UserID and password, which is clearly intended to defeat brute force attacks. However, there seems to be no upper limit on the number of subsequent incorrect login attempts that are made, so a patient hacker with information about a user might be able to keep continually trying. Motor Tax Online and Amazon have no apparent policy for handling unsuccessful login attempts because 25 incorrect combinations were entered on both of them with no consequences.

Bank of Ireland stood out as an exception in that, when users were logging in, they were clearly notified of the time and date of the last successful login on that account. With Gmail, there is a feature to check the last activity but this is not prominent, and none of the other systems featured similar notices. The relevance of clearly letting a user know when an account was last accessed is that the user is made aware of any illegitimate activity that may have taken place.

4.3 Analysis of Procedures for Changing and Recovering Passwords

It is noteworthy that of the twelve Websites evaluated, not one compels a user to *ever* change his password. The author ashamedly admits that he retained a weak password, now altered, for more than 10 years on one of these sites. If a user is not regularly forced to change his password, he is likely to mislay it [31]. All of the systems surveyed have a mechanism for recovering lost passwords/PINs, though in the cases of *Bank of Ireland 365online* and *PAYE anytime* this necessitates off-line postal correspondance. For those systems that permit a user to change or recover a password on-line, the processes vary. If a user knows his password and is logged in, all that is normally required is to go to the account settings and specify the old and new passwords and perhaps one or two other authenticating facts; this is to prevent somebody else changing your password. In cases where a user does not know his password and cannot log in to this account, he is generally required to authenticate himself by providing certain facts or answers to questions that were configured during the account registration process. Once these facts and questions are correctly answered, what generally happens is that a confirmation reset code is sent to the designated email address of the user. Additionally, a number of sites use CAPTCHAs at various points in the interaction workflow so as to prevent automated agents trying to intercept password recovery.

However, there are a number of potential security risks in the way that user authentication is designed in this type of password recovery process. Firstly, many of the so-called "secret" questions actually pertain to facts which are likely to already be known to others or are readily discoverable, such as a telephone number, date of birth, postcode, or mother's name. Other questions may not be known but can be guessed with a reasonable likelihood of success, such as PayPal's question about one's favourite cartoon character (apparently Bugs Bunny is the usual suspect, as evidenced by numerous Internet polls). Some of the sites provide obscure question sets (e.g. Gmail gives the options of your frequent flyer number, first telephone number, first teacher's name, or library card number), which leaves the user with no option but to nominate his own secret question. The dummy user that was created in this study chose to exercise this latter option and suggested that his secret question should be "What is your favourite film?" to which he responded, being a regular sort of fellow, "The Shawshank Redemption". The point is that when users are given the option of nominating their own questions, there is a strong chance that they will choose a predictable question with a predictable answer.

The second problem with this general type of password recovery process is that, like any system of control checks and balances, it is only as robust as its weakest link. The user's email account is a key factor in the process, being the channel through which changes to other accounts are notified and verified, so the implicit assumption is that the user's email account is secure. However, if a user's email account were to be broken into, an impostor could change settings in that user's other accounts and clear his tracks by deleting all the confirmation messages, meaning that the user may not find out until sometime later that one or more of his accounts was misused. Given that many people now habitually use a suite of Websites (e.g. Gmail/Facebook/Twitter/YouTube, eBay/PayPal) there is a risk of a domino effect occurring if they all, as is not uncommon, share the same UserID and password.

Finally, in relation to resetting passwords, the only site amongst those evaluated which prevents the reuse of all former passwords is eBay. The reuse of passwords is a security hazard because many people like to recycle the same combinations again and again; this is akin to handing someone a bunch of keys and saying "I don't know which one it is, but if you try all of these you'll get in".

4.4 Policies and Information for Users

It is very notable that on many of the Websites evaluated (e.g. Gmail, Amazon, LinkedIn, eBay), the link to the Privacy Policy is inconspicuously hidden within the page footer in a place that most users would rarely notice or care to look. The ethics of this type of interface design are questionnable; it almost appears as though organisations are intentionally keeping their privacy policy out of clear sight of customers. The Bank of Ireland 365online Website takes a very different approach, being exemplary in the visibility and clarity of information that it provides in relation to privacy, security, and fraud prevention. Contrary to the approach taken by the aforementioned examples, Bank of Ireland has very prominent links on its entry level pages directing customers to useful and intelligible sources of guidance. The other bank evaluated, Rabobank, provides substantially less information as regards advice for users, preferring instead to proclaim its "100% No-Fraud On-Line Banking Guarantee".

PayPal provide an obvious link to a Security Center where users can obtain advice against phishing, spoofing, and identity theft. It is interesting however that the PayPal site permits a user to enter a very weak password (see Table 2) which is in direct contravention of the recommendation in their own Identity Theft Guide to "always choose strong passwords to protect accounts; mix upper and lowercase letters; use symbol characters". Unlike PayPal, the link on eBay's Website to their Safety Center is buried in the small print. Confusingly, the "Search" button in eBay only searches auction listings; there is no obvious way of searching or browsing eBay pages, so users are forced to really drill-down or use an external Internet search engine in order to locate information about fraud prevention on eBay's site.

5 Conclusions

Many of the design principles outlined in the literature review of this paper appear to be self-evident, yet the findings of this research indicate that most of the Websites analysed do not enforce simple validation rules on password selection and take a rather lax approach to privacy assurance. At a time when Internet criminals are devising ever more sophisticated schemes to defraud on-line users, it would seem that a lot of Websites are "leaving the key under the mat" by not taking adequate precautions to safeguard UserIDs and passwords from being violated. Of particular importance is the need to lock down email accounts with strong passwords because if an email account is violated it can be abused as a launchpad to gain access to further accounts, thus raising the likelihood of identity theft and financial fraud. It is quite absurd to think that a common document like a Curriculum Vitae may be all that a wily "social engineer" requires to initiate an elaborate scam.

To the extent that security is considered in the literature on information systems development, the main focus has been on the software engineering of secure protocols and services, security modelling, and programming aspects. There is comparatively little work on the human-centred aspects of security and their implications for user interaction design, data validation rules, and process logic. One of the mantras of user interaction design is "don't make me think!", meaning that a system should be designed in such a way that it is obvious to users what to so. However, as the findings of previous studies [7, 12, 31] reveal, users don't seem to think about the probability or consequence of their account being violated, most likely because it is not obvious to the average user what the risks are or what to do to mitigate those risks. It is therefore important to take pre-emptive action during systems analysis and design by anticipating the mistakes that users are likely to make, which requires the requirements analyst to be aware of how the system is actually going to be used and potentially abused or misused. Norman [17] makes the point that "developers who lack an understanding of real human behavior tend to impose logical rules". Software engineers and programmers have traditionally taken an inside-out approach to systems development, concentrating primarily on the internal back-end components. On the other hand, interaction designers take an outside-in approach, concentrating on the user's front-end experience with a particular emphasis on usability and the (mis)conduct of users. The principle of the "path of least resistance" tells us that a user will take convenient shortcuts around the intended course of action if he is not prevented from doing so. This means that in order to develop a secure information system, the back-end and front-end designers must work together to devise solutions which are cognisant of behavioural tendencies that potentially compromise security. For example, most "password strength meters" currently work on the assumption of a brute force attack, but in actuality it is usually much easier to crack a password using a personal information attack. Therefore, additional intelligence should be designed into the logic of database validation rules so as to also disallow elementary passwords based on information stored in a user's profile.

Although security and usability can sometimes be regarded as getting in each other's way in the design of Web-based information systems, what they have in common is that historically they were largely ignored until the latter stages of the traditional systems development lifecycle but are both now recognised as being foremost design issues that must be considered from the outset [29]. Security design is now being incorporated into the newer generation of systems development methods and modelling techniques [19, 21], and indeed methods from previous generations can also be used for security analysis and design [2]. Thus to conclude, what is now needed to progress the current state of knowledge is to bring together researchers and practitioners from the fields of user interaction design, security engineering, psychology, organisational behaviour, and Internet criminology so as to gain a multi-faceted understanding of the individual, organisational, and societal aspects that need to be considered when designing secure user-centred information systems.

References

1. Alexander C (1964) Notes on the synthesis of form. Harvard University Press, Cambridge
2. Baskerville R (1993) Information systems security design methods: implications for information systems development. ACM Comput Surv 25(4):375-414
3. Cranor LF, Garfinkel S (2005) Security and usability: designing secure systems that people can use. O'Reilly Media. ISBN 10: 0-596-00827-9
4. Data Protection Commissioner (Ireland) (2008) Twentieth annual report of the data protection commissioner. http://www.dataprotection.ie. Accessed 20 Apr 2010
5. Egger FN (2001) Affective design of e-commerce user interfaces: how to maximise perceived trustworthiness. In: Helander HM et al (eds) Proceedings of the international conference on affective human factors design. Asean Academic Press, London
6. Fernandez-Buglioni E, Hybertson D, Sommerlad P (2005) Security patterns: integrating security and systems engineering. Wiley, New York
7. Furnell S (2007) An assessment of website password practices. Comput Secur 26(7–8):445–451
8. Gao H, Liu X, Wang S, Dai R (2009) A new graphical password scheme against spyware by using CAPTCHA. In: Proceedings of the symposium on usable privacy and security (SOUPS), 15–17 July, Mountain View
9. Halkidis ST, Chatzigeorgiou A, Stephanides G (2006) A qualitative analysis of software security patterns. Comput Secur 25(5):379–392
10. Internet Crime Complaint Centre (2009) Internet crime report. http://www.ic3.gov/media/annualreports.aspx
11. Keith M, Shao B, Steinbart PJ (2007) The usability of passphrases for authentication: an empirical field study. Int J Hum Comput Stud 65(1):17–28
12. Lang M, Devitt J, Kelly S, Kinneen A, O'Malley J, Prunty D (2009) Social networking and personal data security: a study of attitudes and public awareness in Ireland. In: Proceedings of the international conference on management of e-commerce and e-government (ICMeCG), Nanchang, China, 16–19 Sept. IEEE CompSoc, pp 486–489
13. Lyardet F, Rossi G, Schwabe D (1999) Discovering and using design patterns in the WWW. Multimedia Tools Appl 8(3):293–308
14. McDermott J, Fox C (1999) Using abuse case models for security requirements analysis. In: Proceedings of 15th annual computer security applications conference (ACSAC'99), Phoenix
15. Microsoft (2005) Web service security: scenarios, patterns, and implementation guidance for web services enhancements (WSE) 3.0. http://msdn.microsoft.com/en-us/library/aa480569.aspx. Accessed 20 Apr 2010
16. Muoz-Arteaga J, Gonzàlez RM, Vanderdonckt J (2008) A classification of security feedback design patterns for interactive web applications. In: Proceedings of 3rd international conference on internet monitoring and protection, IEEE Computer Society, Washington, pp 166–171
17. Norman D (2009) When security gets in the way. ACM Interact 16(6):60–69
18. Preece J, Rogers Y, Sharp H (2002) Interaction design: beyond human-computer interaction. Wiley, New York
19. Schneier B (2004) Customers, passwords, and Web sites. IEEE Secur Privacy 2(4):88
20. Sindre G, Opdahl AL (2005) Eliciting security requirements with misuse cases. Req Eng 10(1):34–44
21. Siponen M, Heikka J (2008) Do secure information system design methods provide adequate modeling support? Inf Softw Technol 50(9–10):1035–1053
22. van Duyne DK, Landay JA, Hong JI (2003) The design of sites—patterns, principles and processes for creating a customer-centered web experience. Addison-Wesley. ISBN 020172149X
23. Vora P (2009) Web application design patterns. Morgan Kaufmann. ISBN 012374265X
24. Vu K-PL, Proctor RW, Bhargav-Spantzel A, Tai B-L, Cook J, Schultz EE (2007) Improving password security and memorability to protect personal and organizational information. Int J Hum Comput Stud 65(8):744–757

25. Wadlow T, Gorelik V (2009) What can be done to make browsers secure while preserving their usability? Commun ACM 52(5):40–45
26. Yan J, Blackwell A, Anderson R, Grant A (2004) Password memorability and security: empirical results. IEEE Secur Privacy 2(5):25–31
27. Yapp P (2001) Passwords: use and abuse. Comput Fraud Secur 9(1):14–16
28. Yee K-P (2002) User interaction design for secure systems. In: Deng R et al (eds) Information and communications security, LNCS 2513. Springer, New York, pp 278–290
29. Yoder J, Barcalow J (1997) Architectural patterns for enabling application security. In: Proceedings of 4th conference on pattern languages of programming (PLoP'97), Monticello, 3–5 Sept
30. Zurko ME, Simon RT (1996) User-centred security. In: Proceedings of new security paradigms workshop, Lake Arrowhead, ACM Press, New York, pp 27–33. ISBN 0-89791-944-0
31. Zviran M, Haga WJ (1999) Password security: an empirical study. J Manage Inf Syst 15(4):161–185

A Framework for Intention-Driven Requirements Engineering of Innovative Software Products

Mauri Leppänen, Juha Lamminen and Pertti Saariluoma

Abstract Requirements engineering is highly challenging particularly when designing innovative software products. This is so because there are no corresponding products, ultimate needs of actors are difficult to capture, the products may have unforeseeable impacts on the actors' behavior, and it is hard to find out how value-added and competitive the product actually is. In this paper, we propose a novel framework for intention-driven requirements engineering of innovative software products, which combines technological, social and business viewpoints. We illustrate its use with a short example related to the domain of web mapping services and augmented reality.

1 Introduction

Requirements engineering (RE) is the most challenging discipline in the systems development lifecycle. Requirements are often ambiguous, incomplete, redundant and contradictory, due to stakeholders' divergent perspectives, terminology, and interests [13]. They are frequently changing because stakeholders are not able to say what they really need. Requirements engineering also involves reluctant participation, misperception and disagreement. Many RE languages and frameworks have been proposed to help making requirements more precise, complete, and consistent [2, 19]. These techniques are mainly targeted to late-phase requirements engineering, and less attention has been given to consider how the desired software product would meet personal or business goals, and why the product is needed.

The emergence of novel IC technologies has created new possibilities to add value through innovative products and services. Realizing this potential requires creativity

M. Leppänen (✉) · J. Lamminen · P. Saariluoma
Department of Computer Science and Information Systems, University of Jyväskylä,
P.O. Box 35 (Agora), 40014, Jyväskylä, Finland
e-mail: mauri@cs.jyu.fi

J. Lamminen
e-mail: juha.e.lamminen@jyu.fi

P. Saariluoma
e-mail: pertti.saariluoma@jyu.fi

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_33,

and innovative acts across the whole product design lifecycle. *Innovation* means a multi-stage process whereby organizations transform ideas into new/improved products, service or processes, in order to advance, compete and differentiate themselves successfully in their marketplace [1].

Requirements engineering is particularly challenging when it concerns the design of an innovative software product. First, there is no corresponding product from which to learn. Second, a new product may affect, in an unforeseen manner, ways of how actors behave and business is run. Third, it is hard to discover how value-added and competitive the product-to-be would actually be, and for whom, among the networked companies. These challenges have been tried to be met by moving the focus onto early phase RE [30, 31], by developing goal-oriented approaches [3, 5, 14], and by crafting creativity techniques [8, 21] for requirements engineering. Nevertheless, the situation is still unsatisfying.

We argue that in RE of innovative products creativity should be a built-in, all-bracing property, the intentions of human and business actors have to be taken as the basis for all the RE activities, and the way of requirements engineering should be tailored based on the situation at hand (cf. customer-driven vs. market-driven vs. technology-driven). We suggest a novel framework for RE of innovative software products, which aims to satisfy the abovementioned demands. The framework combines the technological, social and business viewpoints. It is aimed to be used for analyzing and comparing existing RE methods in terms of how they address, emphasize, and integrate creativeness and the three viewpoints. It also provides a basis for considerations of how to enhance existing RE methods.

The paper is organized as follows. In Sect. 2 we describe the framework, and in Sect. 3 we illustrate it with a small example. In Sect. 4, a short literature review of relevant literature is presented. Section 5 concludes with the summary.

2 Framework for Requirements Engineering

We define *requirements engineering* (RE) as a creative process in which stakeholders and designers work together to create and concretize ideas for a new product [17, 18]. We build our RE framework on four concepts: innovation, user-centressnes, goal-orientedness, and multi-viewpoint. First, the overall structure of the framework is designed to boost the capturing and elicitation of new ideas for software products. Second, the RE process starts with finding out the intentions and needs of users, and users are expected to have an active role in the RE process. Third, goals provide rationale for requirements that operationalize them and help detect and manage conflicts among the requirements [14]. Fourth, requirements engineering is considered from the business, social, and technology viewpoints. The focus of the first viewpoint is on value creation, distribution and consumption (cf. [7]). Social viewpoint concerns members of communities and their social relationships. Technology viewpoint is applied to bring forward and discuss new technological innovations.

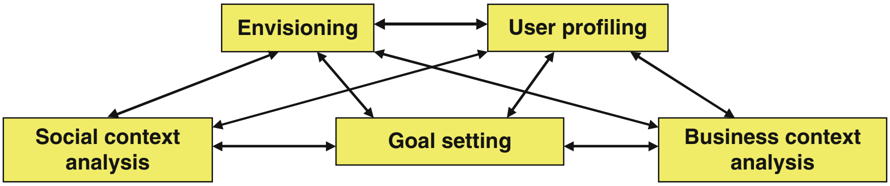


Fig. 1 Framework for requirements engineering

The RE framework is composed of five interrelated RE activities: envisioning, user profiling, business context analysis, social context analysis, and goal setting (see Fig. 1). In the following, the activities are described in more detail.

2.1 Envisioning

Envisioning means innovative and creative action aiming at evoking and eliciting ideas for new products and/or new ways of using the products. Product design often starts with a vague desire, concept, or image of something new. Work is done, through innovative techniques [8, 19], to elaborate it towards a more concrete vision. To ensure its feasibility, the vision should be shared within the community and be translatable into reality. Creativity is a key factor to successful envisioning. It is the interplay between the ability and process by which an individual or group produces a product that is both novel and useful within some social or business context [24]. *Technical innovations* improve existing features or facilitate the introduction of completely new features. *Need innovations* happen when a hidden need is found or an existing need is abstracted into a more profound need (cf. [11]). *Business innovations* help companies increase their profits through the use of a new product and by re-engineering their business processes.

Envisioning can be driven by technology, business, or human needs. In the first case, a new IT intensive product is first invented and then an attempt is made to find out contexts in which its features could be utilized [15]. In the second case, the focus is on searching "enablers" for innovative business solutions [31]. In the third case, a social context is analyzed to understand what profound needs the human actors have, and to figure out how the needs could be satisfied through a new product. Our RE framework support all of these approaches. There is a large range of creative models and techniques available for envisioning [8, 21].

2.2 User Profiling

User profiling aims at distinguishing user categories and characterizing them in terms of relevant features. Typically, the features include psychological characteristics (e.g., attitude, motivation), knowledge and experience (e.g. typing skill, task experience), job and task characteristics (e.g. frequency of use), and physical characteristics (e.g. color blindness) [20]. Characteristics can be concretized and combined by attaching them to personas, or fictional people [4]. Early personas are sketches that are later elaborated toward more detailed characters. Personas should not replace active user participation.

Data for user profiling is collected through interviews and/or user profile questionnaires, and indirectly from marketing personnel [19]. As new products built upon emerging technologies seldom have existing counterparts, it is important also to gather information about the users' subjective opinions and preconceptions regarding their future (cf. cultural probes [6]).

2.3 Business Context Analysis

If the product is to act as an "enabler" for innovative business solutions, not just as a means of automating well-established business processes, one has to obtain a deep understanding about the domain. This means learning about the interests, priorities and abilities of various business players**.** Also, to ensure that the idea of a new product will really add value, it is necessary to analyze its profitability. The purpose of *business context analysis* is to model the context, objectives, and processes of the business entity for which a new software product is to be designed, in order to better understand its ICT needs and potentials. A business context means a web of networked companies and/or organizations that are established to provide goods and/or services to consumers.

There are several approaches and models for business context analysis (e.g., business process models, workflow models, cultural models). We do not propose any new approach, but utilize two of them that are particularly suitable for early-phase requirements engineering. They are the e^3 value approach [7] and the i* framework [30–32].

In the e^3 value approach [7], a business context is viewed from a commercial perspective. It is seen as a multi-actor network in which economic value is created, distributed and consumed through a software product. An innovative idea means to find new economic value propositions that are yet unknown to the market and significantly change the way a company does business [7]. The e^3 value approach, as it is integrated into our RE framework, starts from an innovative idea resulting from envisioning. Based on this, a baseline model is constructed. The model shows business actors, value activities and value objects, as well as concepts related to value exchange through the distribution of value objects among the business actors [7]. Construction of the model may yield more new ideas which can be iteratively elaborated through envisioning and perhaps through social context analysis. Based on the improved understanding of the business context and a clearer conception of the economic value of a new innovative idea, it is possible elicit goal statements for the product and its use.

In the i* framework [32], early requirements are assumed to involve actors who depend on each other for goals to be achieved, task to be performed, and resources to be furnished. The goals are analyzed and elaborated into functional and non-functional requirements of the product-to-be. The key concept is actor. Organizational actors are viewed as having intentional properties such as goals, beliefs, abilities and commitments. Actors are strategic in the sense that they are concerned about opportunities and vulnerabilities, and seek rearrangements of their environments that would better serve their interests [32]. The i* framework includes the strategic dependency (SD) model and the strategic rational (SR) model. The former is used for describing the dependency relationship among various actors in the business context. The latter is used to describe stakeholder interests and concerns, and how they might be addressed by various configurations of products and environments [32].

Both of these approaches emphasize the importance of understanding the business context, motivations and rationales (the "Whys"). The way of executing business context analysis depends on the situation at hand. For instance, if economic value is important, an e^3 value model is first constructed. The approaches can also be used to analyze the "eco-systems" of competitors and combine the results with information got from market analysis, in order to contemplate the competitiveness of the product.

2.4 Social Context Analysis

The purpose of *social context analysis* is to make sense of the motives and actions of the members in a given social community. A *social community* means a group of people who share common characteristics or interest and is perceived or perceiving itself as distinct in some respects from the larger society with which it exists [28]. Communities can be established based on family or friendship relationships, ideological views, hobbies, work, etc. Novel IC technologies (e.g. Facebook, Twitter, LinkedIn, Bebo, and MySpace) have substantially helped establishing new communities and networking within and between them.

Resulted from the first iterations in envisioning and user profiling, there exist preliminary conceptions about the relevant contexts and actors. Here, these conceptions are elaborated and analyzed. Within a family, for instance, the actors are the father, the mother, and children, and possibly grand parents and other relatives [9]. The next step is to recognize the intentions of the human actors. The purpose is to analyze the underlying rationale and purpose of what people are doing: what are they trying to achieve, and why are they trying to achieve it?

During social context analysis, problems are uncovered and analyzed to find out whether ICT in some form could support social actions. The analysis may result in more elaborated ideas on a desired product, or it may lead to the conclusion that no product is needed. Another approach is to start with considering how an existing technology, perhaps in a new form and/or in a novel manner, could be utilized by the community. In parallel to the work in goal setting, work here continues with describing actions the actors are doing to reach their intentions and goals. Features of the product are outlined, and user tasks are modeled and analyzed to help discover main functionalities and qualities of the product. There are various models and techniques that can be used to describe user tasks in the social context: e.g., scenarios, (essential) use cases, use case templates, task decomposition trees, and work flow models.

2.5 Goal Setting

Goal setting means the activity by which human and business actors' intentions are captured and refined into goals and ultimately specific, preferably measurable, requirements for a new product. An *intention* is a mental state of the actor, which motivates and regulates actions [2]. During this process, the goals and requirements are discussed, negotiated, formalized and prioritized. At its best, the requirements become concise, feasible, precise, complete, consistent and verifiable. As our focus is on early requirements engineering, we do not discuss formalization.

A *goal* is the reason for which something is done, made, used etc. Goals can be formulated at different levels of abstraction, ranging from high-level, motivational and strategic concerns to low-level, technical concerns [13]. From the business viewpoint, goals are objectives of the business organization, which guide decisions at various levels. From the human actor's viewpoint, goals are conditions or states of affairs that the actor would like to achieve. The lower-level goals concern objectives related to the concrete use of a product. A *requirement* specifies properties (functional, structural, physical, etc.) of the product-to-be [16]. A *feature* is a property of the product. It is first a design feature, then an implementation feature, and finally a usage feature (see Fig. 2).

Goals are elicited, elaborated and analyzed in parallel to social context analysis and business context analysis. If the product is aimed for a social community, intentions and actions of the members give the starting point for elicitation. If the product is to be designed for business, business goals of and relationships between the business

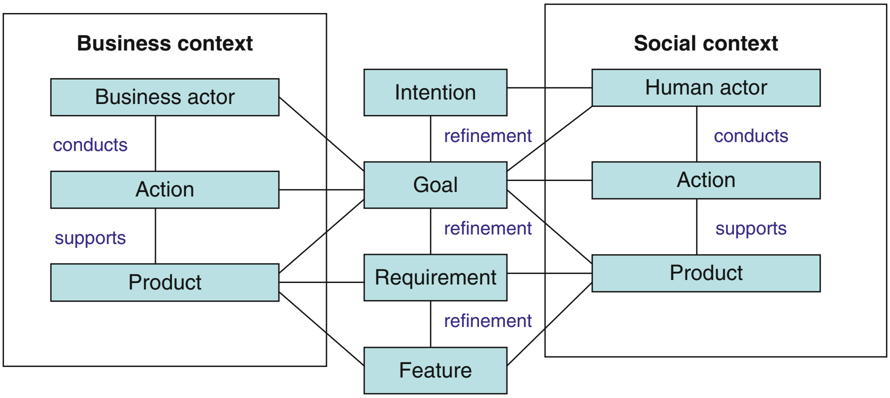


Fig. 2 Main concepts of the framework

actors provide a basis for goal elicitation. In case the product has some features analogical to existing products, problems and deficiencies encountered in their use can be negated and used as anti-goals for the product-to-be. Goals can also be got from marketing groups, technical support groups, and derived from competitive analysis.

Once goals have been identified, they are refined progressively into lower-level goals until they involve the use of the product. This process is done by asking the HOW questions. Another approach is to derive more abstract goals from those already identified by asking WHY these goals exist. This way it may be possible to refine them and find their sub-goals that were originally left undetected [14]. The goals and requirements can be described in AND/OR trees or networks. As all the goals and requirements cannot be satisfied, some priorization is necessary (for priorization techniques see [10]).

3 Example

In this section, we illustrate the RE framework with a hypothetical example. Let us suppose that we are designing a new product to help tourists and start with the following vague ideas: planning trips should be fun, experiences during trips should be enriched, and networking with and communication between people with the same interests should be encouraging. The product is to be built upon three novel technologies: location-based services (LBS), distributed geographic information systems (GIS), and augmented reality (AR).

Most LBS offer basic navigation and routing services (cf. GoogleLatitude). We are especially interested in web map services (cf. Google Maps, Yahoo! Maps, Microsoft Virtual Earth, MapQuest, ArcWeb) in the form of distributed GIS that enhance the accessibility and dissemination of geographical capabilities and knowledge to internet users [27]. AR facilitates overlying labels or other virtual information over the real world objects (e.g. scene, building) observed through a camera attached to a computer, thus helping information contextualization and localization [29].

In the following, we describe how the activities of our framework can be applied to innovate the novel software product. Our case mainly applies technology-driven approach to RE. Due to the space limit, the description is very indicative. Intentions and goals are not discussed separately (cf. Goal setting) but mentioned in connection with the other activities.

Envisioning. We seek innovative solutions from three perspectives: technology, social and business. For finding technology innovations, we first recognize that there already exists a wide range of web mapping products. Our task is to invent a product with new features which make it desirable and competitive. We decide to build upon three promising concepts: geoportal, online annotation editing, and collaboration support. *Geoportals* are websites providing entries to geographical content on the Web [26]. Through them it is possible to provide tourists with a large assortment of existing web map functionalities and services [27]. Today, only a few augmented reality (AR) applications allow the *online creation and editing of annotations* [29].

When AR becomes popular on mainstream devices, there will be a large group of users who will be able to add content. This potential we want to exploit. We believe that with online annotation techniques an explosion of user created content, similar to the increase of online context with Web 2.0 applications, will be experienced [29]. Thereupon, the emergence of new forms of *social networking and cooperation* related to traveling can be also expected.

User profiling. The World Tourism Organization defines tourists as persons who "travel to and stay in places outside their usual environment for more than twenty-four (24) hours [. . .] for leisure, business and other purposes [. . .]". A tourist wants to obtain new experiences (e.g. roads with hairpin bends), learn exotic cultures or new things, or raise his/her social status (cf. visits in glamorous places). Tourists differ from each other in pre-knowledge, motivational, psychological, educational and physical terms. These differences should be taken into account in the functionality and interaction design of the product-to-be.

Social context analysis. We consider the context of traveling in three stages: trip planning, travelling and retrospective recalling and discussions of the experiences. The way a trip is planned depends on the motives of a human actor. Here, we assume that tourists are mainly interested in attractions. They want to know, among others, which kinds of places and attractions there are, how to reach them, where to have lunch and accommodation. Trip planning should also be fun, not stressful as it is usually. In addition to searching information related to geographical objects through common web service functionalities, tourists want to make their personalized maps and/or geotags. Because travelers plan less and less ahead of a trip and engage in more on-route and in-destination planning, there is a need to support "ad-hoc decisions" on restaurants or attractions. Tourists want also to be active members of social communities. Therefore, they should be facilitated to search for other travelers with similar profiles, interests and travel experiences, and their personalized maps (social bookmarking) [26].

During the trip, a tourist needs many kinds of guidance, not only that provided by GPS navigators. To offer a tourist richer experiences from places and attractions, the product should provide virtual information, attached with objects seen in the real world. This way, they can "see", for instance, how the building that is now in ruins looked like in the past. Tourists also want alternative viewpoints from which to look at objects around them. For example, it should be possible to select "eyeglasses" through which (s)he is provided with virtual information related to architecture, history, sport, or culinary art in the certain place. Tourists want also to include location related items (hotels, attractions, monuments, restaurants) on their personal maps and enrich them with feedback and experience of the places.

To refresh memories and share experiences with people, the product should allow to create new networks based on geographical routes and location-items (map networking [26]), and to use them for communication on interesting topics.

Business context analysis. The product connects a wide range of stakeholders, including tourists and their social networks, tourist bureaus, restaurants, pizzerias, hotels, museums, art galleries, etc. In this multi-actor network, economic value is created, distributed and consumed through the software product [7]. There is a large variety of ways of how tourism companies can exploit a product like ours in their business [26]: localized exploitation, internal integration, business process redesign, business network redesign, and business scope redefinition. Companies can generate greater business benefits when they increase their level of exploitation. The product-to-be should provide a large range of options for exploitation. Next, we only give some examples of them.

Map networking transforms a trip planning process to a more collaborative and social decision making process where social networks with other travelers are incorporated into value chains and become co-creators and co-producers of travel services [26]. A hotel's website does not only allow customers to search for, contribute and read user-generated content, but the hotel can also use this content for developing its new services [26]. Virtual information attached to places and attractions in web maps enables a tourism business update their information even on an hourly basis and provide, for example, a special deal for the day if the restaurant is short of customers or a hotel is having a low occupancy rate [23].

To recognize the involved stakeholders and to examine their complex value-added relationships, we should build an e^3 model [7]. The model shows, among others, key tourism business actors, value activities and value objects. In addition, to elaborate goals, beliefs, abilities and commitments of the business actors and strategic dependencies between them, we may find the strategic dependency (SD) model and the strategic rational (SR) model [31] useful.

Concluding from business context analysis, we can state that the product-to-be should provide tourist companies with new opportunities to redesign their operations, internally and/or externally, and to develop new collaborative business models by involving new partners and/or users-customers into their value chains and systems.

4 Related Work

There are a large number of studies on innovation and creativity in requirements engineering (RE). Nguen and Shanks [22] present a theoretical framework for understanding creativity in RE. Kauppinen et al. [11] identify three main opportunities for innovations: discovering hidden user and customer needs, inventing new product features, and supporting feature development. Maiden et al. [18] present a scenario-driven RE process, called RESCUE, that integrates human activity modeling, system goal modeling, and creativity techniques. Maiden et al. [17] apply theories from cognitive science to build creative models and working methods in air traffic management domain. Grube et al. [8] propose a framework to select creativity techniques for requirements elicitation. Gordijn et al. [7] consider RE from business perspective and distinguish three stakeholder-type related viewpoints: value, business process and information system viewpoints.

There is also a wide range of studies on goal-oriented approach to RE. Chung et al. [3] present a goal-based framework for clarifying and prioritizing non-functional requirements. The KAOS methodology [5] contains a rich set of formal analysis techniques and three types of models: goal model, object model, and operation model. i* [30, 31] is an agent-oriented modeling framework that supports the modeling activities before the system requirements are formulated. Shibaoka et al. [25] proposes a method called GOORE which deploys a domain ontology to support goal decomposition. There are also some goal-oriented requirements engineering methods, such as AWARE, AGORA, and Tropos.

We have strongly exploited earlier research on innovation to make creativity an all-bracing property of the framework. Our approach borrows ideas from goal-oriented approaches but goes further by emphasizing the importance of the human actors' intentions as the starting point for requirements engineering. Business context analysis has been built on [7] and [30, 31]. The RE research has also yielded multiple generic conceptual and functional frameworks for requirements engineering, such as those by IEEE, ISO and [12]. These are, however, mainly lifecycle-based, whereas our framework is focus-based. It shows which you should particularly focus on in requirements engineering.

In summary, our framework integrates creativity and the derivation of requirements from the intentions of human actors (cf. social context) and the goals of business actors (cf. business context).

5 Summary and Conclusions

We are experiencing challenging times in today's software product design. Real business potential is only gained from radically novel products, not from making improvements in existing products. A growing portion of products is directed to so-called consumer markets (cf. social media software, game industry), for which it is typical that nobody knows how a new product should be like. In this situation, it is not enough to ask what is needed but why it is needed. This can only be done by deriving requirements from the intentions of human and business actors.

We have suggested a new intention-driven framework which combines the technological, social and business viewpoints for requirements engineering of innovative software products. It is composed of five activities: envisioning, user profiling, social context analysis, business context analysis, and goal setting. The framework can be used to analyze and compare existing RE methods in terms of how they address and integrate creativeness, user-centredness, business view, and goal-orientness. It also provides a basis for considerations of how to enhance existing RE methods, in order to make them better meet today's challenges. The framework equally applies to software product design for consumer markets as well as for company use. In the former, social context analysis is emphasized whereas in the latter business context and social context (within a company) are evenly important.

Our next step is to elaborate the framework to address, in more detail, interaction requirements engineering, especially regarding user experience, which is important to social media and other novel application domains. It is also important to pay attention to the so-called micro-innovation processes, i.e., thought processes, concepts and design thinking paradigms, which actually create new ideas of the products, and develop innovation techniques based on them. Thirdly, future research is needed to tailor the activities of the framework to be part of agile methods whose popularity is growing fast in practice.

References

1. Baregheh A, Rowley J, Sambrook S (2009) Towards a multidisciplinary definition of innovation. Manage Decis 47(8):1323–1339
2. Bratman M (1987) Intentions, plans, and practical reasons. Harvard University Press, Cambridge
3. Chung L, Nixon B, Yu E, Mylopoulos J (2000) Non-functional requirements in software engineering. Kluwer, Boston
4. Cooper A (1999) The inmates are running the asylum. Macmillan, New York
5. Dardenne A, van Lamsweerde A, Fickas S (1993) Goal-directed requirements acquisition. Sci Comput Prog 20(1–2):3–50
6. Gaver B, Dunne T, Pacenti E (1999) Design: cultural probes. Interactions 6(1):21–29
7. Gordijn J, Akkermans J (2003) Value-based requirements engineering: exploring innovative e-commerce ideas. Req Eng 8(2):114–134
8. Grube P, Schmid K (2008) Selecting creativity techniques for innovative requirements engineering. In: Proceedings of the workshop on multimedia and enjoyable requirements engineering (MERE'08), pp 32–36
9. Hughes R Jr, Hans J (2001) Computers, the internet, and families: a review of the role new technology plays in family life. J Fam Issues 22(6):778–792
10. Karlsson L, Höst M, Regnell B (2006) Evaluating the practical use of different measurement scales in requirements prioritisation. In: International symposium on empirical software engineering (ISESE 2006), pp 326–335
11. Kauppinen M, Savolainen J, Männistö T (2007) Requirements engineering as a driver for innovations. In: IEEE Requirements Engineering Conference, pp 15–20
12. Kotonya G, Sommerville I (1998) Requirements engineering, processes and techniques. Wiley, Chichester
13. van Lamsweerde A (2009) Requirements engineering—from system goals to UML models to software specifications. Wiley, Chichester
14. Lapouchnian A (2005) Goal-oriented requirements engineering: an overview of the current research. Depth Report, University of Toronto
15. Leffingwell D (2007) Scaling software agility—best practices for large enterprises. Addison-Wesley, Boston
16. Lin J, Fox M, Bilgic T (1996) A requirements ontology for engineering design. Concurr Eng 4:279–291
17. Maiden N, Gizikis A, Robertson S (2004) Provoking creativity: imagine what your requirements could be like. IEEE Softw 21(5):68–75
18. Maiden N, Manning S, Robertson S, Greenwood J (2004) Integrating creativity workshops into structured requirements processes. In: Designing interactive systems (DIS2004), pp 113–122
19. Maiden N, Rugg G (1996) ACRE: selecting methods for requirements acquisition. Softw Eng J 11(3):183–192
20. Mayhew D (1999) The usability engineering lifecycle: a practitioner's handbook for user interface design. Morgan Kaufmann, San Francisco
21. McFadzean E (1998) The creativity continuum: towards classification of creative problem solving techniques. J Creativity Innov Manage 7(3):131–139
22. Nguyen L, Shanks G (2009) A framework for understanding creativity in requirements engineering. Inf Softw Technol 51(3):655–662

23. Pan B, Crotts J, Muller B (2007) Developing web-based tourist information tools using google map. In: Sigala M, Mich L, Murphy J (eds) Information and communication technologies in tourism. Springer, Vienna, pp 503–512
24. Plucker J (2003) Creativity. College of Arts and Sciences at Indiana University, Bloomington
25. Shibaoka M, Kaiya H, Saeki M (2007) GOORE: goal-oriented and ontology driven requiremetns elicitation method. In: Hainaut J-L et al (eds) ER workshops 2007, LNCS 4802. Springer, New York, pp 225–234
26. Sigala M, Marinidis D (2009) Exploring the transformation of tourism firms' operations and business models through the use of Web map services. European and Mediterranean conference on information systems (EMCIS2009), Izmir
27. Tait M (2005) Implementing geoportals: applications of distributed GIS. Comput Environ Urban Syst 29:33–47
28. Webster 1989. Webster's encyclopedic unabridged dictionary of the english language. Cramercy Books, New York
29. Wither J, diVerdi S, Höllerer T (2009). Annotation in outdoor augmented reality. Comput Graphics 33(6):679–689
30. Yu E (1997) Towards modeling and reasoning support for early-phase requirements engineering. In: Proceedings of the IEEE symposium on requirements engineering, pp 226–235
31. Yu E (1999) Strategic modeling for enterprise integration. In: Proceeding of the 14th world congress of the international federation of automatic control, Beijing, China
32. Yu E, Mylopoulos J (1996) AI models for business process reengineering. IEEE Exp 11:16–23

Using Real-Time Event Stream Framework to Develop RFID-based Retailer Supermarket Systems

Guoqiong Liao, William Wei Song, Lei Shu and Changxuan Wan

Abstract With the increasing mature of hardware technology and the continuous reduction of RFID production cost, the Radio Frequency Identification (RFID) technology has been widely used in many fields. This paper proposes a real-time event stream framework for RFID system and develops a retail supermarket management system for the purpose of commodity monitoring and tracing. An event-centric architecture for the RFID real-time event stream processing is designed, which can support both real-time event queries over event streams and history event queries on event databases. Event types of retail supermarkets are analyzed in details, and the event relation diagram among them is given. Based on the framework, an event filtering method using a hash table to remove redundant data and an improved SASE method to process complex events are presented.

1 Introduction

Radio Frequency Identification (RFID) is an automatic identification technology, and has become an important part of Internet of Things (IOT). Comparing with the traditional identification technologies, the RFID technology has many advantages. For example, it can collect a large amount of information automatically without human intervention, and can identify multiple tags simultaneously. In recent years, with the rapid advancement of the hardware technology and the continuous reduction of the RFID production cost, the RFID technology has been widely used in many fields, which require real-time monitoring and tracking, such as retail industry, logistics and supply chain, library, equipment and asset management [3, 4, 6, 10].

Because the RFID technology can help to reduce cost, improve efficiency, make shopping activities more convenient, and prevent theft, retail supermarkets begin to use it for sale and commodity monitoring. For example, Wal-Mart has used electronic product code (EPC) to track commodities in logistics; TESCO in the United Kingdom

W. W. Song (✉)
School of Engineering and Computing Sciences, Durham University, Durham, United Kingdom
e-mail: w.w.song@durham.ac.uk

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_34, © Springer Science+Business Media, LLC 2011

has developed RFID-enhanced shelves; PRADA, a clothing chain vendor in Italy, has equipped their dressing rooms with the RFID techniques; Estee Lauder, a skin-care product company, has applied the RFID technology for selling men's specific skin-care products. Obviously more and more enterprises are adopting the RFID-based technology for their retail supermarket/shops where each single commodity is attached with a RFID tag.

In order to improve the management efficiency for product identifications in retail supermarket, it is unavoidable for each product to be attached with a RFID tag, and deploy large-scale RFID readers around the supermarket. However, widely adopting this technology will lead to problems discussed as follows:

- Large data volume. Large-scale deployment of RFID readers will produce huge amounts of data. According to statistic results, the data generated by Wal-Mart within three days have been more than all the data that U.S. National Library contains. Even a company with a middle-scale RFID deployment generates more than one billion bytes of data each day.
- Data redundancy. The data set generated by RFID readers may contain a large number of redundant data due to following reasons: (1) a reader may read the same tag many times within a reading cycle; (2) multiple adjacent readers may read the same tag simultaneously; (3) because of signal reflection, a reader may detect the tags that are not in its detection range (i.e., false positive) [5].
- Real-time processing. By exchanging RF signals, a RFID reader can capture tagged objects around it. If the reading cycle is small (e.g., several millisecond), the continuously arriving data form a data stream, thus it is necessary to process them in real-time way. Otherwise, a lot of useful information may be lost [1, 14].
- Semantic information. The tagged objects observed by RFID readers usually carry a lot of implicit semantic information about the context status and background knowledge of objects. For example, the position of a tagged object can be learned from the location of the readers detecting it; if an object is observed by the reader staying in a cashier, it means the item is being sold, and the number of stock of the commodity should be reduced by 1. So it is necessary to extract high-level semantic information from the raw data.

Therefore, how to filter the redundant raw RFID data effectively, process data stream in real time, and dig out the hidden semantic information to obtain more accurate and meaningful information for higher level applications, have become an urgent research issue in development of RFID application systems.

By now, none of the existed RFID systems have met these requirements mentioned above. The key issues to design and implement such system include:

- Establishing a real-time event streams processing architecture;
- Supporting both real-time event query over event streams and history event query on event databases;
- Defining original event streams and complex event streams;
- Processing the events in an efficient way, including original event filtering and complex event matching.

In the research area of RFID middleware, more and more attentions have been focused on the RFID event handling mechanism, such as event definition language, event model, complex event processing strategies, etc. [7–9, 13–15]. However, these studies still remain at the theoretic investigation stage, and few of their research results have been applied in real systems. In addition, some methods are too complicated to be used in dealing with real-time data streams.

In [11], a design of an event-driven RFID-based system for retail supermarkets is proposed, but it lacks any real-time processing ability, and does not strongly support event stream processing. Based on the work, this paper will extend event-driven model in a real-time event stream processing environment. The main contributions of the paper are listed as follows:

- The architecture for RFID real-time event stream processing is designed, to support both real-time event query over event streams and history event query on event databases.
- The event types of retail supermarket are analyzed in details, including three kinds of original events and six kinds of complex events, and an event relational diagram is given.
- An efficient event filtering method to remove redundant data and an improved SASE method to process complex events are presented.

The remaining of this paper is organized as follows. Section 2 is the framework of RFID real-time event stream processing. Section 3 analyzes the event types of retail supermarkets and gives the event relation diagram among these events. In Sect. 4, original event definition languages are designed, and an event filtering method is suggested. Section 5 defines complex event stream, and gives its matching procedure. The last section concludes the paper.

2 Framework of RFID Real-Time Event Stream Processing

An event usually represents that an action or a state change occurs in the real world. "Event-driven" means that data is encapsulated in an event and handled as an event. In RFID systems, an observation of a reader detecting a tag can be regarded as an original event. However, these original events may contain a large number of redundant data. If these redundant data do not be filtered, they will greatly increase the overhead on network transformation and data processing, and even lead to making wrong decisions. Therefore, filtering and processing the original events has become one of the most important issues in the RFID systems.

Definition 1 An observation of a reader detecting a tag is called an *original event.*

An original event can be expressed as a triple: <RID, OID, Timestamp>, where, RID is the reader's ID, OID is the tag's ID, and Timestamp is the time when the event happens.

Definition 2 An event which is derived from one or more other events according to event matching rules is called a *complex event*.

A complex event can be expressed as a binary: <Element, Rule>, where, Element is a set of either original events or complex events, and Rule represents a set of operation rules.

The framework of RFID event stream processing is shown as Fig. 1.

- User Interface: providing interfaces for users or subscribers to define events (EDSL), query real-time events (SASE) and history events (SQL);
- Event Capturer: capturing original events reported by readers;

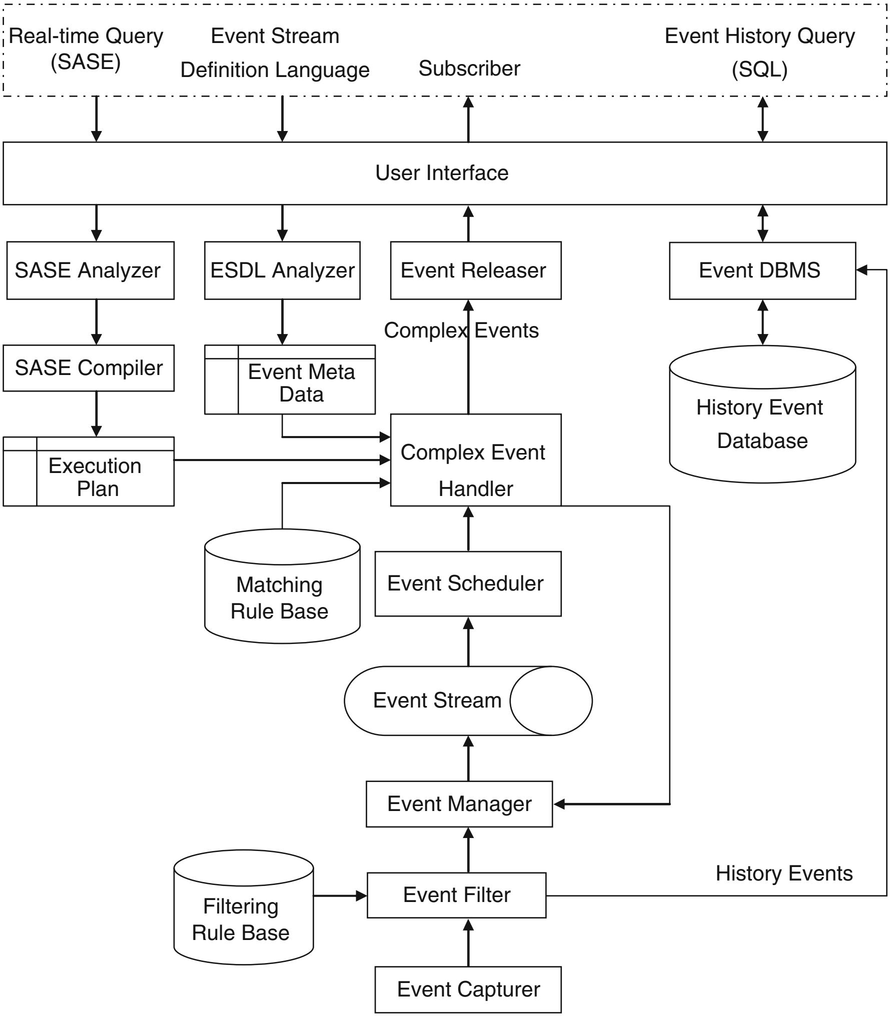


Fig. 1 Framework of RFID real-time event stream processing

- Event Filter: filtering events according to the rules storing in the filtering rule base;
- Event Scheduler: scheduling events to process according to their priorities, which are assigned by EDF (earliest deadline first) policy;
- Complex Event Handler: matching the original and complex events in the event stream to generate high-level complex events;
- Event Manager: organizing events according to their timestamps, and sending them to the event stream; remove out-of-date events to History Event Database;
- Event Releaser: publishing event information according to the requirements of subscribers.

3 Event Classification and Relation Diagram in Retail Supermarkets

Based on the framework discussed above, we have implemented an event-driven RFID-based retailer supermarket simulation system, which provides the following main It functions:

- Commodity monitoring. It can monitor the real-time statuses of all commodities, and automatically report abnormal behaviors of commodities, such as misplacing, theft, out-of-stock, etc.
- Location query and history tracing of location changes. It can query current location of specific commodity in the supermarket, and can trace the history of location changes of any lost commodity.

The general business procedures of a retail supermarket can be described as follows:

1. After entering the supermarket, a customer first fetches a shopping cart, and then goes to the shopping area for selecting commodities;
2. During shopping, the customer may move a commodity to the shelf which is not its pointed shelf. That is, a misplace event occurs.
3. At the end of shopping, when the customer passes the cashier, a checkout event is triggered. The system will check whether the remaining quantity of the commodity below a predefined threshold, to decide whether to trigger an out-of-stock event.
4. If a customer taking a commodity leaves the supermarket without payment, a stolen event will be triggered.

Therefore, the system takes into account three kinds of original events and six kinds of complex events. The original events include:

- Shelf-reading events (SR): original observation events from the readers deployed in shelves;
- Counter-reading events (CR): original observation events from the readers deployed in counter;

- Entrance-reading events (ER): original observation events from the readers deployed in the entrances.

The Complex events include:

1. On-shelf event (NS): means that a tag is on the shelf it should be put on.
2. Loss event (LS): means that an unsold commodity can not be detected by any readers in the supermarket.
3. Misplace events (MP): means that a reader in a shelf detects the commodities belonging to other shelves.
4. Payment events (PA): if a commodity is observed by the reader in the counters, a payment event is triggered.
5. Out-of-stock events (US): means that the inventory level of a commodity is lower than a threshold.
6. Stolen events (ST): means that a commodity is stolen.

The relational diagram of these events can be shown in Fig. 2.

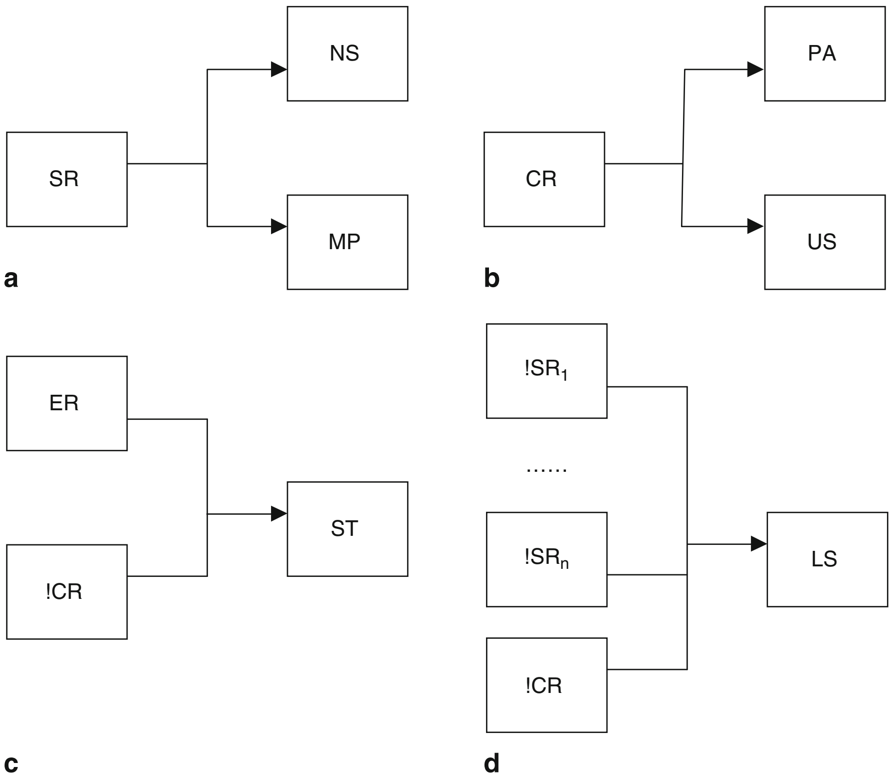


Fig. 2 Event relation diagram

4 Original Events Definition and Filtering

The definition of the original events should support continuous queries over data streams. The statement to define an original event stream is [2, 9]:

```
CREATE STREAM primitive-event AS
SELECT TagID, ReaderID, timespan(timestamp, interval), interval
FROM Reader_Reading
GROUP BY TagID, ReaderID
```

For example, the schema of Shelf-reading event is SR(EventID, ReaderID, TagID, Timestamp), which can be defined as follows.

```
CREATE STREAM SR AS
SELECT TagID, ReaderID, timespan (timestamp, interval), interval
FROM ShelfReader_Reading
GROUP BY TagID, ReaderID
```

This paper proposes a simple hash table-based filtering strategy. The basic idea is to use a hash table to store original event flows. Each tagged object is assigned a unique identifier OID, as a key in the hash table.

When a new original event <RID, OID, Timestamp> arrives, the filter will check whether the OID is in the hash table. If it is, that means the tag already exists in the current window. At this time, only the event with lager timestamp remains in the event stream. If it is not, then the new event is written into the event stream.

```
DuplicateEventFilter (size)
Begin
    EventTable← emptyHashTable;
      loop{
         Event← the next happened Event;
         oldEvent← EventTable[Event.OID];
         if(oldEvent=null){
             EventTable.put(Event.OID, Event);
             output Event;
        }
        else if (Event. timeStamp-oldEvent. timeStamp>size){
              output Event;
              oldEvent. timeStamp← Event. Timestamp;
           }
             else
              oldEvent. timeStamp← Event. timestamp;
      } //end loop
    End
```

5 Complex Event Definition, Matching and Query

5.1 Complex Event Definition and Matching

The complex event stream is generated through the continuous queries for the basic event streams. Its syntax is as follows [2, 9]:

CREATE STREAM complex-event AS
SELECT TagID, ReaderID, [other attributions]
FROM basic-event
GROUP BY TagID, ReaderID

For example, the schema of Misplace events is MP(EventID, ReaderID, TagID, Timestamp, PreLocationID), which can be defined as follows.

CREATE STREAM MP AS
SELECT ReaderID, TagID, Timestamp, PreLocationID
FROM Shelf_Reading
GROUP BY TagID, PreLocation

As an expression language supporting complex event queries, SASE is an advanced RFID complex event processing method [7]. It can support negative operation (¬), parameterized predicates, sliding windows, etc. However, SASE needs a large memory to store stack data, especially for non-events. To overcome this shortcoming for SASE, this paper designs an optimized complex event detection method based on SASE, using the finite automaton (NFA) and hash table together.

Firstly, a complex event SEQ (A, ..., F) must meet the following conditions:

1. a_i.OID=...=f_i.OID
2. a_i.Timestamp<...< f_i.Timestamp
3. f_i.Timestamp—a_i.Timestamp $\le$ sliding windows

The complex event matching procedure includes two steps: complex event detection and complex event output.

1. *Complex Event Detection*

Firstly, a complex event handler gets an input event from the event stream, and determines whether the event meets some matching rules. If the event is an initial event of NFA or there are some precursor events of the event in the hash table, then store it in the hash table. If the event is the last one of the NFA, then output a complex event. In particular, if the input event is a non-event, all precursor events of the event will be deleted from the hash table, in order to save memory space.

```
ComplexEventDetection ( )
Begin
    generate NFA according to the configuration file;
    while(!end( ))
   {
     get one event (e) from the input event stream;
     if (NFA(e)!=0)
        continue; // The event is a unrelated one.
     else
     {
      if(e is a non-event)
         delete all relevant events of e's precursor events from hash table;
      else
       {
        if (e is the first event of NFA)
            keep e in the hash table;
       else
       {
            if (e's precursor events contains e)
            {
               if (e is the end event)
                  ComplexEventOutput ( ) //output the complex event;
               else
                  keep e in hash table;
            }
       }
     }
   }//end while
END
```

2. *Complex Event output*

```
ComplexEventOutput ( )
   Begin
     pCurrentEvent= the pointer of end event;
     While (pCurrentEvent!=NULL)
     {
        Push_stack(pCurrentEvent);
        if (pCurrentEvent precursor events contains the event)
          {
            Push_stack(pCurrentEvent precursor events pointer);
            pCurrentEvent=pCurrentEvent precursor events pointer;
          }
        else
            return;
        pCurrentEvent=pCurrentEvent precursor events pointer;
     }//end while
```

```
While (stack()!=NULL)
   {
      Pop_Stack()
      Output the complex event;
   }
END
```

5.2 Query Complex Event Based on SASE

SASE is a high-level language structure, similar to SQL. Its structure is as follows [7]:

[FROM <stream name>]
EVENT <event pattern>
[WHERE <qualification>]
[WITHIN <window>]
[RETURN <return event pattern>]

- FROM clause provides the name of the input stream. If it is omitted, the query will use a default input.
- EVENT, WHERE and WITHIN clause forms the event pattern matching block. EVENT clause specifies the input stream to match the event pattern.
- WHERE clause, if there is, means restrictions based on value will be added on the events which are declared in model.
- WITHIN clause specifies a sliding window in the event mode. Event matching block transforms the input stream into a new composite event stream.
- RETURN clause transforms the composite event into the final output. This conversion can choose a subset of attributes and can also calculate an aggregate value similar with the SELECT clause in SQL. It can give name for the output stream and event type; furthermore, it can call database operations to retrieve and update data.

We can use SASE to query complex events

1. Query the location of a given tag

   EVENT SR x
   WHERE x.TagID='TagID'
   RETURN RetrieveLocation (x.TagID)

2. Query TID, name and location of the misplace commodity

   EVENT SEQ (SR x, SR y, ! (ANY (CR, SR) z))
   WHERE [TagID] $\wedge$ x.ShelfID $\neq$ y.ShelfID $\wedge$ x.shelf_id = z.ShelfID
   WITHIN 1 hour

RETURN x.TagID, x.ProductName, RetrieveLocation (z.TagID)

3. Query TID, name and location of the stolen commodity

   EVENT SEQ (SR x, !(CR y), ER z)
   WHERE x.TagID=y.TagID ∧ x.TagID=z.TagID
   WITHIN 12 h
   RETURN x.TagID, x.ProductName, RetrieveLocation (x.TagID)

4. Query name, quantity and the number of the out-of-stock commodity

   EVENT SEQ + (SR x)
   WHERE [ProductName] ∧![ID]
   WITHIN 5 Seconds
   HAVING Count (x.ProductName) < Threshold
   RETURN x.ProductName, Count (x. ProductName)

5. Query TagID, name, last seen time and location of the lost commodities

   EVENT SEQ (SR x, !(ANY (CR, SR) y))
   WHERE [TagID] ∧ x.ShelfID = y.ShelfID
   WITHIN 1 hour
   RETURN x.TagID, x.ProductName, RetrieveLocation (x.TagID)

6. Query sales of top 10

   EVENT SEQ + (CR x)
   WHERE [ProductName] ∧![ID]
   WITHIN 15 Minutes
   GROUP BY x.ProductName
   ORDER BY Count (x.ProductName)
   RETURN Top-10(x.ProductName, Count(x.ProductName))

6 Conclusions

How to develop RFID-based systems efficiently and find the valuable information accurately and timely remains a big challenge issue for the RFID system developers. This paper proposes a framework for RFID real-time event stream processing, as well as presents the event filtering and event matching methods, to meet the requirements of RFID-based monitoring and tracing applications. Although this work is done on the background of retail supermarket applications, it is easy to extend it in other applications.

For the next step, we plan to carry on this research in the following three aspects: (1) to design an effective real-time event scheduling algorithm by taking into account of the characteristics of RFID event processing; (2) to develop a scenario-sensitive monitoring model based on the contextual semantic description model [12] which provides a complete conceptual representation of various objects in the domain of interest; and (3) to improve the suggested event filtering method with more factors such as objects on transportation and their inner structures using a deep-structure analysis of Supporting Vector Machine (DS-SVM).

References

1. Agrawal J, Diao Y, Gyllstrom D, Immerman N (2008) Efficient pattern matching over event streams. In: Proceedings of ACM international conference on interaction sciences: information technology, Wilderness, South Africa, pp 147–160
2. Bai Y, Wang F, Liu P, Zaniolo C, Liu S (2007) RFID data processing with a data stream query language. In: Proceedings of 23rd international conference on data engineering (ICDE). Istanbul, Turkey, pp 1184–1193
3. Brusey J, Harrison M, Floerkemeier Ch, Fletcher M (2003) Reasoning about uncertainty in location identification with RFID. In: Workshop on reasoning with uncertainty in robotics at proceedings of the 18th international joint conference on artificial intelligence, Acapulca, Mexico, July
4. Chawathe SS, Krishnamurthy V, Ramachandran S, Sarma S (2004) Managing RFID data. In: Proceeding of 30th VLDB conference, Toronto, Canada, pp 1189–1195
5. Floerkmeier Ch (2004) A probabilistic approach to address uncertainty of RFID, Auto-ID Labs Research Workshop, Zurich, Switzerland
6. Fuhrer P, Guinard D, Liechti O (2006) RFID: from concepts to concrete implementation. In: Proceedings of the international conference on advances in the internet, processing, systems and interdisciplinary research, 30th Apr
7. Gyllstrom D, Wu E, Chae HJ, Diao Y, Stahlberg P, Anderson G (2007) SASE: complex event processing over streams. In: Proceedings of conference on innovative data system research, Asilomar, California, USA, pp 108–119
8. Hinze A (2003) Efficient filtering of composite events. In: Proceedings of the British National Database Conference, London, UK, pp 207–225
9. Jin X, Lee X, Kong N, Yan B (2008) Efficient complex event processing over RFID data stream. In: Proceeding of 7th IEEE/ACIS international conference on computer an information science, Portland, Oregon, USA, pp 75–81
10. Khoussainova N, Balazinska M, Suciu D (2007) Probabilistic RFID data management. In: UW CSE technical report. http://db.cs.washington.edu/peex/papers/UW-CSE-07-03-01.pdf
11. Shu L, Liao G (2009) Event-driven RFID system development for retailer supermarket. In: Proceedings of first international conference on information science and engineering, Nanjing, China, pp 2316–2319
12. Song W (2006) A semantic modelling approach to automatic services analysis and composition. In: The IASTED international conference on web technologies, applications, and services (WTAS), Calgary, Canada, July
13. Wang F, Liu S, Liu P, Bai Y (2006) Bridging physical and virtual worlds: complex event processing for RFID data streams. In: Proceedings of international conference on extending database technology (EDBT), Munich, Germany, pp 588–607
14. Wu E, Diao Y, Rizvi S (2006) High-performance complex event processing over streams. In: Proceedings of ACM SIGMOD international conference on management of data, Chicago, Illinois, USA, pp 407–418
15. Xiao J, Yuan W, Wang D (2009) Complex event processing model for distributed RFID network. In: Proceedings of ACM international conference on interaction sciences: information technology, culture and Human. Seoul, Korea, pp 1219–1222. http://portal.acm.org/toc.cfm?id=1655925&type=proceeding&coll=GUIDE&dl=GUIDE&CFID=87820888&CFTOKEN=60912819

Having a Customer Focus in Agile Software Development

Garry Lohan, Michael Lang and Kieran Conboy

Abstract This research looks at the customer focus of agile software development teams. The study is part of a larger study examining how the twelve principles of Beyond Budgeting are operationalised in the context of an agile development environment. Using two case study sites and a semi-structured interview approach the customer focus of agile teams operating within two large organisations is examined. In these organisations the direct customer is not the end user of the product; rather they are another group within the organisation downstream of the agile development team. The results suggest that while organisations may espouse to have a customer focus the structures may not be in place to enable sufficient sharing of customer knowledge and utilisation of customer feedback. Emergent themes from the study suggest that customer identification, customer characteristics, customer location and the teams' experience of the customer and their domain may have an impact on the customer focus of an agile team.

1 Introduction

The importance of the customer to agile software development teams operating within large organisations (Augustine 2005; Highsmith 2004) and to the organisation as a whole (Gulati 2007; Gulati and Oldroyd 2005) is well documented. The construct "customer focus" has been developed by the Total Quality Management (TQM) literature (Ahire et al. 1996; Sousa 2003). Ahire et al. (1996) developed and validated a customer focus construct in the context of manufacturing firms. They found that quality is influenced by top management's commitment through customer focus. Issac et al. (2004) developed a conceptual framework for TQM in software organisations which also included the customer focus construct and included client (customer) feedback and client involvement as part of the construct dimensions. While various dimensions of the customer focus construct have been studied in the Information

G. Lohan (✉)
National University of Ireland, Galway, Ireland
e-mail: Garry.Lohan@nuigalway.ie

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_35,

Systems (IS) community, such as customer communication or relationships (Korkala et al. 2009), customer satisfaction (Mann and Maurer 2005) and customer involvement (Kautz 2009), the only research in IS found on the customer focus construct as validated in the TQM literature is from Ravichandran and Rai (2000) who used the end user participation as representative of the customer. However in agile software development the customer may be represented by actual customers, customer proxies, product managers or product champions (Highsmith 2004). There is a gap in the literature surrounding the development of the customer focus construct when the customer is not the actual end user, but rather another group within the organisation.

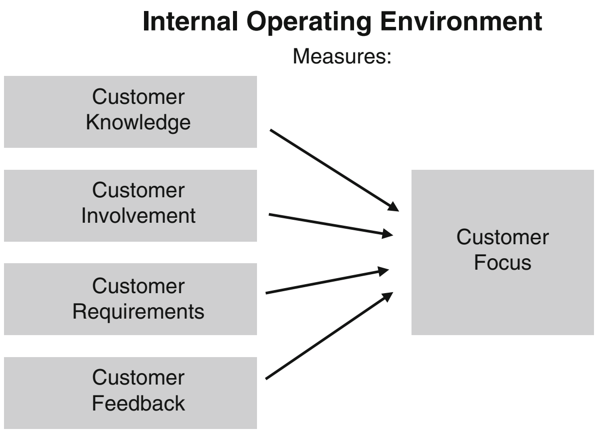
This research looks at the customer focus of agile teams within two large organisations where access to the actual end users may be problematic. In these organisations the customer is internal and the customer focus construct is examined from the perspective of the agile team delivering products to an internal customer. This is an increasingly important area as more and more large organisations begin using agile methods. To date there has been little research carried out on the customer focus of an agile team delivering products to an internal customer. In research within IS in general, there has been a tendency to focus only on specific aspects of customer focus as opposed to adopting a more holistic approach. This research hopes to address these issues by introducing and developing a customer focus construct through case study research.

The next section of this paper outlines the theoretical development of the customer focus construct and introduces the conceptual framework. Section three introduces the research sites and research methodology. Section four highlights the findings and section five discusses emergent factors coming from the research. Section six is a discussion with a revised framework and finally section seven concludes with implications for industry and research.

2 Theoretical Development

Customer focus practices involve the establishment of links between customer needs and satisfaction and internal processes (Sousa 2003). The quality management literature has developed instruments to measure the customer focus of the organisations quality management program (Sousa 2003; Ahire et al. 1996; Flynn et al. 1994). These instruments have been conceptualized and contextualized within an IS environment (Issac et al. 2004) and have been adapted for use in an IS environment where the end user replaced the role of the customer (Ravichandran and Rai 1999). Other fields such as organisational literature have also developed ways to measure and utilize customer focus (Gulati 2007; Gulati and Oldroyd 2005). With the introduction of agile software development processes, the role of the customer and the customer focus of the agile team take on a new significance. This research draws on previous research on the customer focus construct. Through a review of the customer focus construct dimensions discussed in the TQM literature we have developed our conceptual framework for this study (Fig. 1). We have adapted the construct dimensions from the previous studies to examine the customer knowledge, customer

Fig. 1 Conceptual framework



involvement, customer requirements and customer feedback loops of agile software development teams.

2.1 *Customer Knowledge*

Gulati and Oldroyd (2005) suggest a four stage process for understanding the customer focus journey. The first stage is the collection of *information* on customers. This is then consolidated and analysed to gain an insight into customers from past behaviour. This insight is then used to develop a likely understanding of future behaviour, which is used to provide real-time responses to customer needs. Customer satisfaction surveys are the standard way to gather information and gain an insight into customer perceptions. Surveys not only give general information on customers but also information on what customers are and are not satisfied with. To achieve the level of coordination and cooperation required from a customer focused organization, the correct *structural mechanisms, processes and incentives* need to be in place. These will allow employees to focus on the customer by harmonizing information and activities across units, and by encouraging people in all parts of the company to work together in the interest of customer needs. Sharing this customer knowledge is also critical in utilizing the cognitive resources within a team (Srivastava et al. 2006).

2.2 *Customer Involvement*

Taking the already created construct dimension from the quality management literature this research defines customer *involvement* as the extent of the customers' involvement in the product design process. Sousa (2003) describes the customer focus construct in terms of establishing strong *relationships* with the customers by emphasizing partnership arrangements and having direct customer contacts (face to face *meetings*, plant visits). The customer is an integral part of the agile team and teams are encouraged to have a collaborative customer-developer relationship which involves a high degree of interaction between the teams and the customer (Highsmith 2004; Beck 2005).

2.3 Customer Requirements

Agile methods rely heavily on inputs from the customer rather than having a predefined set of requirements (Beck 2005; Highsmith 2004). The agile teams are expected to work closely with the customer to gather ongoing requirements throughout the project duration, obtaining timely feedback and information. However, customers' insufficient knowledge of the requirements due to the complexity and size of the system poses significant challenges (Cao et al. 2009). These challenges are even more pronounced when customers are not available or not willing to commit to the project (Fitzgerald et al. 2006).

2.4 Customer Feedback

The mere execution of customer surveys is not useful unless the results are made available to functional areas of the organization (Ahire et al. 1996). Teams should be provided *feedback* on both customer complaints and also on customer satisfaction surveys. This feedback is used for *training* if required and to improve *processes* where needed. Gulati (2007) calls this capability development, and it is a means of ensuring that an organisation has enough people that have the skills to deliver customer-focused solutions and also has the correct processes in place to deliver those solutions.

3 Research Methodology

3.1 Site Selection

Both organisations chosen for this study had within the past few years implemented the scrum methodology into their systems development operations. The development teams within both organisations had traditionally worked with a waterfall development and the transition to agile development processes raised questions on the suitability of the surrounding supporting processes. Organisational structures which had supported the use of the waterfall method meant that emphasis on customer collaboration which is the norm in agile development was a relatively new area for these organisations. In both organisations the end user was not the direct customer of the development team. The teams studied were part of a larger umbrella group and therefore their customers were more often than not an internal downstream function of the organisation.

3.2 Data Sources

In both research sites, data were collected through a variety of methods: unstructured and semi-structured interviewing, document review and observation. Data was collected as part of a larger research project.

The first case study was conducted within the Information Services (IS) division of a large multinational financial consulting firm (FCC) which builds customised software applications for internal clients. The data was collected as part of a larger research project that consisted of an in-depth study conducted over 4 months. Three different scrum projects were studied. Data was collected through on-site observation at iteration meetings and daily scrums, review of documentation, three workshops and nine interviews.

The second study was conducted within the IS division of a large multinational oil and gas firm (SCC) which also builds customised software solutions for internal clients. Here ten interviews were carried out with personnel from four different scrum projects and in three different locations. Follow up phone calls, emails were also used.

3.3 Data Analysis

All transcripts were recorded and transcribed entirely. The transcriptions were imported into NVivo for coding. The data was analysed based on Strauss and Corbin's approach (1998) for open and axial coding, where the initial interview questions and subsequent data analysis was based on the customer focus construct dimensions previously discussed (customer knowledge, customer involvement, customer requirements and customer feedback). Data was initially coded around these four dimensions which provided a list of "seed categories" for initial open coding (Miles and Huberman 1999). During this stage the data collected was divided into the four main categories and then further divided into sub-categories.

The axial coding technique was then used to put the data back together by making connections between the categories and sub-categories. Reflexive remarks and memos made during both the interview stage and the analysis stage helped to interpret the data and lead to the identification of emergent themes not previously considered. Follow up phone calls, emails and site visits were arranged where possible and further documentation obtained when further information was needed or clarification was required. Data collection ended when enough categories had been defined to explain what had been observed at both sites and when no additional data were being collected to develop or add to the categories. At this point, further data collection was unnecessary as the categories were deemed to be "theoretically saturated" (Strauss and Corbin 1998).

Precautions were taken to corroborate the interpretations made (Miles and Huberman 1999; Yin 2003). Emerging categories were checked for representativeness by examining them across participants. For example team members' reports of their experience with their customers were checked against the reports from other team members and the project managers or scrum masters. The participants in the study also provided commentary, correction and elaboration on drafts of the findings and framework.

4 Findings

The customer focus of the organisation is discussed first in relation to the four measures outlined in the theoretical development section of this paper. Then the emergent themes or factors identified during the analysis stage are presented and discussed.

4.1 Customer Knowledge

Customer knowledge involves learning about the customer in order to anticipate future needs and also sharing that knowledge with others within the organisation.

Case A has a number of collaborative websites but none specifically dedicated to the collection and dissemination of customer information. The teams are quite open to sharing but there does not appear to be any formal mechanism or structure in place specifically for customer information. One team member commented that when several tracks were working on the same project there needs to be more knowledge sharing among the teams:

> One thing we could improve on and need to improve on is inter-team or inter-track communication (Team Member)

In case B knowledge gathered and shared on the customer also varied. Sharing customer knowledge was viewed a problematic:

> That is also one of the challenges with so many teams, how do we get interaction across the teams (Project Manager)

In other projects where there was an analysis team and close collaboration with the customer, these were not issues, with one scrum master commenting:

> We have a pretty clear view of what the customer will hopefully need in six months, hopefully (Scrum Master)

4.2 Customer Involvement

Having a good customer relationship involves the customers being directly involved in the development process and the developers having direct contact with the customers via face to face meetings. In case A there was a mixed response to this, some groups had a poor relationship with their customer, while in other projects the relationship appeared to be better:

> There are *some* customers who are really involved, they really know the area and they know the tool (Team Member)

In case B, interactions seemed better with the customer or customer representative. All teams interviewed had good interactions with their customers but this was seen as something that had improved over recent years, rather than a norm:

> We've been lucky, we have got a lot of commitment but it is not a default that everyone is committed (Project Manager)

4.3 Customer Requirements

In case A it was strongly felt that the requirements gathering was an issue for the team members, mainly because of their distance from the customer:

> [speaking about] Their requirements, we didn't even understand. They didn't really make sense. Effectively you are talking to people on this side trying to get an idea of the story rather than going direct to the customer" (Team Member)

One team member actually described getting requirements through a second party as akin to *Chinese whispers.*

In case B the requirements were usually received through the product owner who has a hands-on role in refining the product backlog with the scrum teams. The problem of not getting first hand access to the customer is raised by one developer but in general the requirements are clearly established through face to face meetings and workshops with the customer proxy.

4.4 Customer Feedback

In case A the feedback received was very much at a higher level than direct feedback to the team. Team members agreed that there was little feedback from the customer:

> You'd like a retrospective or you'd like something to say, you know, we didn't need that or you know, the usual… (Team Member)

Case B showed that the team members were involved in the feedback process and work was presented to the customer or product owners on a monthly basis. All team members agreed that there was feedback given from the principle stakeholders in some form:

> People speak clearly about what they are concerned about and what they like (at the sprint review) (Team Member)

5 Emergent Factors Affecting Customer Focus

From the first phase of data analysis it became clear that customer focus fluctuated across cases and indeed across projects. From the subsequent axial coding process four core factors emerged which had an impact on the degree of customer focus within each case and project.

5.1 Customer Identity

The notion of the customer is fundamental to current management paradigms and a major thrust of current programmes of organisational change is to replace management hierarchical control with simulated market control, i.e. organisational departments are defined *as if* customers, and work-colleagues relate to each other as customers (du Gay and Salaman 1992). When the customer is internal within the organisation it is important that their role is clearly defined. In case A, there appeared to be some ambiguity as to the identity of the customer and the role they had to play:

> I guess this other group are our customer, our direct customer (Team Member) *or*
> We could see him [the customer] as the person who asked for this project (Project Manager)

The second case was less ambiguous, the customer was clearly identified as the product owner and they had the role of the customer:

> They are defined as customers (Project Manager)

5.2 Customer Characteristics

Prior research has shown the importance of having the customer collaboration for agile projects (Martin et al. 2004a, b; Kautz 2009; Koskela and Abrahamsson 2004). Koskela and Abrahansson (2004) found that the role of the on-site customer to be very demanding and the customer requires a strong ability to resolve issues rapidly. Martin et al. (2004a, b) found that the customers have a pressured and stressful role. The characteristics of the customer are also highlighted in this study and appear to have a direct impact on the customer focus of the agile team. The following quote from case A illustrate this point:

> I guess there has been a continual struggle on this project, there are ourselves and the proxy customer, those two streams have fallen out of synch (Project Manager)

Case B had differing experiences with the customer, some finding that the customer was apathetic:

> Sometimes they comment on things that are good but often they sort of lean back and get information (Team Member)
> I guess they may have looked at our system in its current state for a couple of hours or so [but] there is not much keen interest there (Scrum Master)

And others finding them very involved:

> [the customer] is actually participating in the demos, in the retrospective meetings, he has been very hands-on and given direct feedback on solutions, what worked, what didn't and so on (Project Manager)

5.3 Customer Location

Agile development values close customer collaboration. The location of the customer is important for communication purposes (Korkala et al. 2009). Korkala et al. found that in distributed teams weak customer relationships may result in inefficient communication. Their findings seem to agree with the findings of this study where an on-site customer was easier to communicate with than a customer not located in the same office. An example from case A is where a team member stated that it:

> Would be nice if there was more synch up between the two groups

And suggested this was due to the fact that the customer was not based on site. An example from case B was where the customer was on site and the communication appeared to be more efficient:

> Sitting on the same floor it was much easier, when they needed help from us they could get it straight away and if we needed clearance or whatever we could go over to them(Team Member)

5.4 Teams Experience with Customer

Relationships are developed over time and the teams experience with the customer and the customer's domain appears to have an impact on the customer focus of the team.

An example from case A is where a customer proxy group who had 3–4 years experience with the customer gave the requirements to the development team. However as the project matured and the team gained experience with the actual customer it:

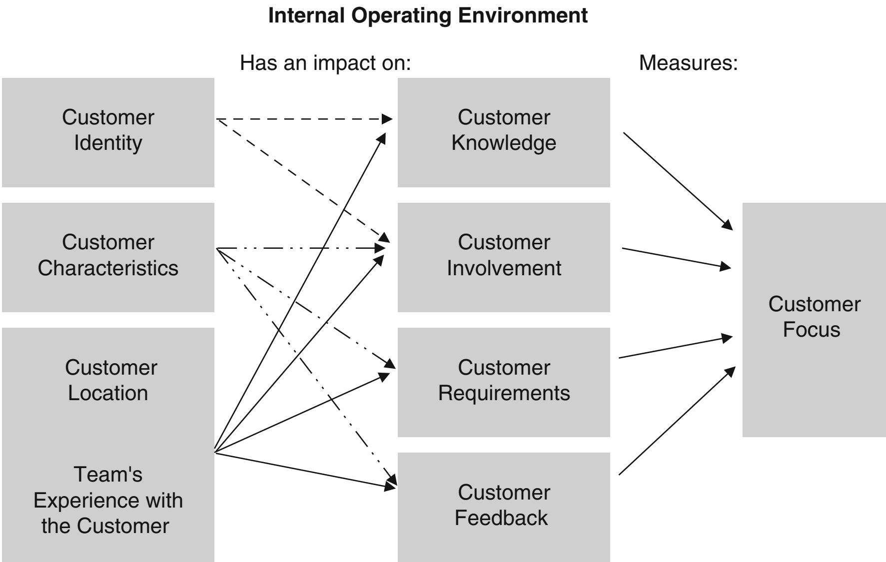


Fig. 2 Revised customer focus framework

> Got to a stage where it was more efficient to deal directly with them and show them what we were building

Showing that as the team gained experience with the customer the relationship improved. An example from case B is where the team:

> Have been involved in the previous product as well so we have a pretty good understanding of the business (Scrum Master)

6 Discussion

The importance of customer focus is evident from the fact that it is one of the six categories among the Malcolm Baldridge Award criteria.[1] The strategic importance of customer focus may vary from organisation to organisation and it should be noted that when developing projects which are for internal customers, strategic priorities and work flow management may impact the relevance of customer focus for any given project. However, customer focus is still one of the vital components of a strong overall performance framework. The revised customer focus framework is presented below (Fig. 2).

The studies presented show a varying degree of customer focus for the agile teams. The initial customer focus construct highlighting customer knowledge, customer relationships, customer requirements and customer feedback does not cover the whole

[1] http://www.baldrige.nist.gov/PDF_files/2009_2010_Business_Nonprofit_Criteria.pdf

	Case A	Case B
Customer Knowledge	No formal structures in place to disseminate customer knowledge, poor inter team knowledge sharing	Inter team knowledge sharing needs inprovement. Scrum of scrums helps disseminate customer knowledge
Customer Involvement	Project dependent	Project dependent
Customer Requirements	Distance from the customer sometimes slowed down the response to requirement queries. Some requirements were not understood	Filtering of requirements through product owner sometimes a problem
Customer Feedback	Little direct formal feedback from customers received	There is an opportunity to get direct feedback at the end of each sprint
Customer Identity	Ambiguity as to the actual customer	Clear structure outlining customer and proxies
Customer Characteristics	This appears to be hit and miss across both cases	
Customer Location	Synchronising across time zones and largedistances was an issue	Customers within same time zone and geographically close.
Teams Experience with the Customer	Both cases experiences improved relationships as the teams experience with the product or the customer increased, indicating a linear relationship between time based experience and improvements in customer relationships	

Fig. 3 Comparison of cases

spectrum of what it means to be a customer focused agile team. Other factors identified highlight the necessity to have a clearly defined customer, the importance of that customer's involvement and attitude towards the team, the location of that customer and the teams working experience of the customer. This produces a refined idea of what having a customer focus is in terms of an agile team producing software for internal customers. More research might take a quantitative approach and examine the links between these construct dimensions and the effects on measurable qualities such as customer satisfaction or customer complaints (Fig. 3).

7 Conclusions

7.1 Importance to Industry

This research highlights has a twofold importance to industry.

It takes the customer focus construct and applies it to the newly emerging agile software development environment. This construct describes the importance of customer knowledge, customer relationships, customer requirements and customer feedback and the two cases studied show how customer focus may be affected within an organisation which develop software systems or applications for internal customers.

The emerging factors show that when an organisation is attempting to create a more customer focused environment they need to consider other factors such as clearly identifying the customer and their role, characterising the customer so the development team can manage their expectations of their customer, being aware that the location and accessibility of the customer impacts customer focus and also the organisation needs to try and establish long lasting relationships between teams and customers.

7.2 Importance to Research

Many organisations have their own internal software development departments which develop products for in house customers. While the concept of customer focus has been researched in other areas, such as manufacturing and also from the viewpoint of the external end user (Ravichandran and Rai 1999), customer focus has not been addressed from the viewpoint of the internal customer. This research is a start to filling this gap and uses the customer focus construct to study two organisations with internal customers. In this sense this research is exploratory and further research could be carried out which includes the emerging factors and compares customer focus across a larger number of organisation contrasting satisfaction ratings across organisations using the extended customer focus construct. The findings show that there are other emerging factors which need to be considered when looking at customer focus, customer identity, customer characteristics, customer location and the teams experience with the customer.

References

Ahire LS, Golhar YD, Waller AM (1996) Development and validation of TQM implementation constructs. Decis Sci 27:23–56

Augustine S (2005) Managing agile projects. Prentice Hall, Upper Saddle River, NJ

Beck K (2005) Extreme programming explained: embrace change. Addison-Wesley, Boston, MA

Cao L, Mohan K, Xu P, Ramesh B (2009) A framework for adapting agile development methodologies. Eur J Inf Syst 18:332–343

Du Gay P, Salaman G (1992) The cult(ure) of the customer. J Manag Stud 29:5

Fitzgerald B, Hartnett G, Conboy K (2006) Customising agile methods to software practices at Intel Shannon. Eur J Inf Syst 15:200–213

Flynn B, Schroeder RG, Sakakibara S (1994) A framework for quality management research and an associated measurement instrument. J Oper Manag 11:4

Gulati R (2007) Silo busting: how to execute on the promise of customer focus. Harv Bus Rev 85: 98–108, 145

Gulati R, Oldroyd JB (2005) The quest for customer focus. Harv Bus Rev 83:92–101

Highsmith J (2004) Agile project management. Addison-Wesley, Boston, MA

Issac G, Rajendran C, Anantharaman RN (2004) A conceptual framework for total quality management in software organizations. Total Q Manag Bus Excell 15:307–344

Kautz K (2009) Customer and user involvement in agile software development. In: Abrahamsson P, Marchesi M, Maurer F (eds) XP2009, Springer, Pula, Sardinia, Italy

Korkala M, Pikkarainen M, Conboy K (2009) Distributed agile development: a case study of customer communication challenges. In: Abrahamsson P, Marchesi M, Maurer F (eds) XP2009. Springer, Pula, Sardinia, Italy

Koskela J, Abrahamsson P (2004) On-site customer in an XP project: empirical results from a case study. Soft Proc Improv Proc 3281:1–11

Mann C, Maurer F (2005) A case study on the impact of scrum on overtime and customer satisfaction. ADC 05. IEEE computer society, Denver

Martin A, Biddle R, Noble J (2004a) On-site customer in an XP project: empirical results from a case study. EuroSPI, Trondheim

Martin A, Biddle R, Noble J (2004b) The XP customer role in practice: three studies. Proceedings of the agile development conference, pp 42–54

Miles M, Huberman A (1999) Qualitative data analysis. Sage, London

Ravichandran T, Rai A (2000) Total quality management in information systems development: key constructs and relationships. J Manag Inf Syst 16:119–155

Sousa R (2003) Linking quality management to manufacturing strategy: an empirical investigation of customer focus practices. J Oper Manag 21:1–18

Srivastava A, Bartol KM, Locke EA (2006) Empowering leadership in management teams: effects on knowledge sharing, efficacy, and performance. Acad Manag J 49:1239–1251

Strauss A, Corbin J (1998) Basics of qualitative research: grounded theory procedures and techniques. Sage, Thousand Oaks, CA

Yin KR (2003) Case study research: design and methods. Sage, Thousand Oaks, California

Ontology-mediated Validation of Software Models

Antonio A. Lopez-Lorca, Ghassan Beydoun, Leon Sterling and Tim Miller

Abstract When errors in software modelling activities propagate to later phases of software development lifecycle, they become costlier to fix and lower the quality of the final product. Early validation of software models can prevent rework and incorrect development non-compliant with client's specification. In this paper we advocate the use of ontologies to validate and improve the quality of software models as they are being developed, at the same time bridging the traditional gap between developers and clients. We propose a general ontology-mediated process to validate software models that can be adapted in a broad range of software development projects. We illustrate this for Multi-Agent Systems (MAS) development providing early evidence of the soundness of our approach. We successfully validate and improve the quality of MAS models for a real-life development project, illustrating the ontology-mediated models validation in a commercial setting.

1 Introduction

Ontologies (understood as a theory about the structure and behavior of the real world in general) provide a mechanism of representing domain knowledge to a varying degree of formalism [4]. They can be utilised by software developers and at the same time read by future users of a system. Our work is in line with what Guarino [8] calls *ontology-driven information system development*. We advocate the use of ontologies to validate and improve the quality of software workproducts during development processes. As an element of joint development with the user, they can bridge common communication gaps between users and developers. We illustrate using an ontology to check consistency, correctness and completeness of models against initial system requirements. We believe that ontologies are generally faster to develop and easier to understand than most analysis and design models that require specific and in-depth methodological knowledge. As an initial system development step, an ontology engineer interviews a client to capture an ontology reflecting their

A. A. Lopez-Lorca (✉)
University of Wollongong, Wollongong, Australia
e-mail: aall645@uowmail.edu.au

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_36,

conceptualisation of their problem and desired features of the solution. We expect that as intermediary modelling elements, ontologies can facilitate and improve the development of software workproducts, potentially reducing the development and maintenance costs of software systems. In this paper, we provide methodology-independent and ontology-based add-on validation processes to facilitate the creation of models for inexperienced modellers and to assist more experienced ones detecting and resolving errors. The ontology can assist modellers throughout the validation processes, which is of particular importance when the domain is complex or not very well known to the modellers. Our proposal is of particular significance for our chosen field of Multi-Agent Systems (MAS). Unlike other disciplines such as Object Oriented Development, MAS development is not so well understood, and due partly to its complexity it has not yet been widely adopted by industry. Whilst the focus of our illustrations is on applying ontologies to improve the development of MAS models, we expect our approach to be easily adaptable to other paradigms such as agile methods.

2 Related Work

The use of ontologies for general software development to validate conceptual models to produce better quality models is not a new idea. However, most existing validation work focuses on using a formal ontology to choose a specific suitable conceptual modelling language for the domain e.g. [14] and more recently [2, 3]. In [2], an Eclipse-based tool is proposed to build and automatically verify conceptual models developed in a language (OntoUML) that uses a foundation ontology to extend UML. In [3], OntoUML conceptual models are automatically transformed to a logic-based language to allow the validation of the modal meta-properties. Our approach is not specific to any modelling language.

Many existing works focus on the use of ontologies to MAS. Of these many focus on the process itself. For example, by designing a reusable ontology allowing complex queries on the domain of "MAS development" in [7] Girardi and her colleagues propose an ontology-based multi-agent development process that can model all the phases of development of MAS. As another example, Nyulas et al. present in [12] an architecture to develop and deploy end-to-end solutions for MAS. They focus on the deployment steps of the system. In [9], a method is given to adapt extreme programming methods to develop a lightweight ontology to help agile development of MAS. It is refined further in [11]. Our focus in this paper is the quality of the workproducts through a domain enriched process rather than the software process itself. Other works use ontologies to assist in the development of workproducts in particular in the detailed design phase. Tran et al. [17] present an ontology-based MAS for the domain of a peer-to-peer (P2P) information sharing community where ontologies are built and used in development-time to create the models and in run-time to exchange information between agents. They use domain ontologies during development and run-time, they do not provide detailed support for the validation of MAS, which is the focus of our proposal. Okouya et al. present [13] a MDA/Ontology approach to improve OperettA, a MAS development framework. They allow the creation of MAS models which are automatically transformed into an ontology. The semantic constraints of the ontology (and of the MAS models) are verified against a MAS domain ontology. They aim to the verification of the models to assess that they have been built properly, but our purpose goes further: we want to validate the models to assess that we have built the correct product according to the user requirements.

Our approach shares similar goals with the work developed by Brandão et al. [6]. They propose the use of ontologies as a method for the verification of MAS designs. They use an ontology to model the MAS modelling language. These model-diagram mappings enable the automatic validation of the models to check that there are neither intra-model nor inter-model inconsistencies. Again, the main difference with our proposal is that they can validate the models against their theoretical structure and dynamics, but use no information about the specification or application domain and their proposal has not been properly validated. Furthermore, they do not generalise their efforts to outside MAS development.

In conclusion, our work uses ontologies to inform modelling of workproducts and is unique in that it is development methodology independent, is focused on the quality of workproducts and does not depend on any specific modelling language. The rest of the paper is structured as follows: Section 3 presents our ontology-based add-on validation process for MAS models and its key features. Section 4 presents a case study in which we apply our process to the simulation of the aircraft turnaround. Section 5 concludes with future work discussion.

3 Ontology-Mediated Add-on Validation Process of MAS Software Models

In this section we present our ontology-mediated process to validate MAS software models. It is important to validate the models as soon as they become available, as the cost associated to errors dramatically increases as the software development process proceeds [18]. Our proposed ontology-mediated MAS software models validation (Fig. 1) consists of five activities that overlap with the development process. Our proposal is an add-on to this core process and completely independent of the underlying software models or their development methodology. Although the model development activity is not in essence part of our proposal, it has been included in Fig. 1 to show that it is intertwined with model validation.

In the *Ontology Development* activity a suitable ontology is retrieved from an existing repository, otherwise one is built using the most suitable ontology engineering techniques. Communication with the client has to be initially intensive to model the domain as detailed and conceptualised by the client. If the ontology lacks details then its effectiveness in the validation and modelling assistance to software developers is reduced. Input to this activity comes through elicitation techniques such as interviewing clients and acquiring any documents that can describe their business processes. For example, in our case study in Sect. 4, in addition to the interviews we

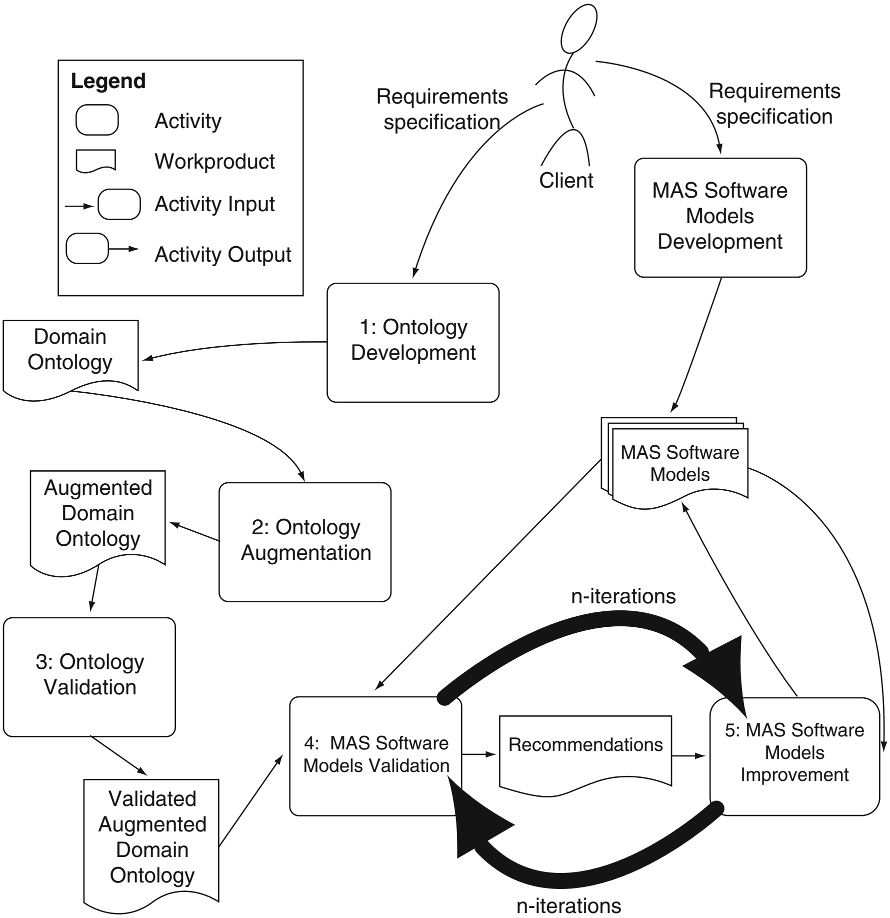


Fig. 1 Ontology-mediated software models validation add-on process overview

use diagrams provided by the client to describe the existing timeline for an aircraft turnaround process.

In the *Ontology Augmentation* activity, the ontology is augmented to represent features related to the chosen development paradigm. Domain concepts are linked to paradigm concepts. Domain concepts are annotated and relations between them are created according existing relations defined for the paradigm. For the paradigm of MAS, we identify terms in MAS modelling: *Goal* (a functional requirement of the system [15]), *Role* (any capacity that the system requires in order to achieve its goal [15]), *Activity* (some work carried out by a role in order to fully or partially fulfil its goal), *Environment* (any entity which is not part of the system but it is needed by the roles to achieve their goals) and *Agent* (a proactive or reactive component of the system plays one or more roles [15]). Some domain concepts are annotated with these

Table 1 MAS-dependent properties used to annotate the ontology

Domain	Property	Range	Domain	Property	Range
Goal	Has a	Goal	Role	Uses	Environment
Role	Responsible for	Goal	Agent	Plays	Role
Role	Participates in	Activity	Activity	Fulfils	Goal
Role	Is peer	Role	Activity	Needs	Environment
Role	Controls	Role	Activity	Precedes	Activity
Role	Is controlled by	Role	Activity	Follows	Activity

terms and related properties are also modelled (summarised in Table 1). Moreover, agents are time-aware. Every decision agents make and every action they carry on has to fit in certain sequence. To specify this sequence, the properties *precedes* and *follows* establish which activities precede and follow which ones.

In the *Ontology Validation* activity, before using the ontology for validating the MAS software models, the ontology itself is validated with the client by various members of the development team. The goal of this is twofold: to ensure that the ontology is compliant and accommodating of the conceptualisation of the client and to secondly ensure a common understanding of the domain across the development team (between persons responsible for developing and for validation).

In the *MAS Software Models Validation* activity, the MAS models are validated against the augmented ontology for consistency and compliance with the client's specification. This activity provides the control element for new iterations. A new iteration will be necessary as long as any recommendation is made to improve the quality of the models. Not all the models can be validated to the same extent using the ontology. Some may be very structured and the use of the ontology will provide specific instructions to improve them. Other models may be composed of free text, for which the use of the ontology will only be able to provide a guideline for the analyst to interpret.

In the *MAS Software Models Improvement* activity, the recommendations are analysed by the developers to choose which to apply and which to ignore. After improving the quality of the MAS models according to chosen recommendations, the new set of models will be used as input for Activity 4 in the next iteration.

Development proceeds with each iteration further along the sequence of workproducts required by the chosen methodology. The development and validation of the MAS software models are intertwined and done concurrently. Problems of reviewed models are fixed before their full development. Any models yet to be commenced in that iteration, will take advantage of the recommendations avoiding compounded errors. The MAS software model development process will follow an iterative, incremental and concurrent development process model.

In order to perform the validation described in Activity 4, the process has to be instantiated: A MAS development methodology has to be chosen and mechanisms to validate the associated MAS models defined.

A recent survey in [16] of ten prominent agent-oriented methodologies shows that there is a set of common models across existing methodologies. The following models are the most common (in increasing acceptance order): Agent model (90%), goal model (60%), interaction model (60%), scenarios (50%), organisation model (40%), role model (30%), and environment model (30%). Without loss of generality, we work with the ROADMAP methodology [10, 15] which provides all those models. Moreover, authors of ROADMAP availed themselves to develop the models for our case study to simulate aircraft turnaround (the process between an aircraft landing at an airport and taking off again). The validation process is based on comparing models and ontology elements pairwise, taking into account their semantics. For example, suppose that the relation `Aircraft transports Luggage` is defined in the ontology, while in the environment model it is stated that `Aircraft carries Baggage`. Both are equivalent in our domain.

As an example, we show the mechanisms to validate the two more popular models of the ROADMAP methodology, the goal and the agent model.

A *Goal Model* can be seen as a use case for an open and distributed system [5, 15]. It sub-divides the main goal of the system into sub-goals and specifies roles participating in the fulfilment of each goal (e.g. Fig. 2). The ontology can ensure that all the specified goals are accounted for, the roles integrity and hierarchy is maintained. The goal model validation consists of the following proposals:

1. To add to the model any roles defined in the ontology but not used in the goal model, and removing those not defined in the ontology.

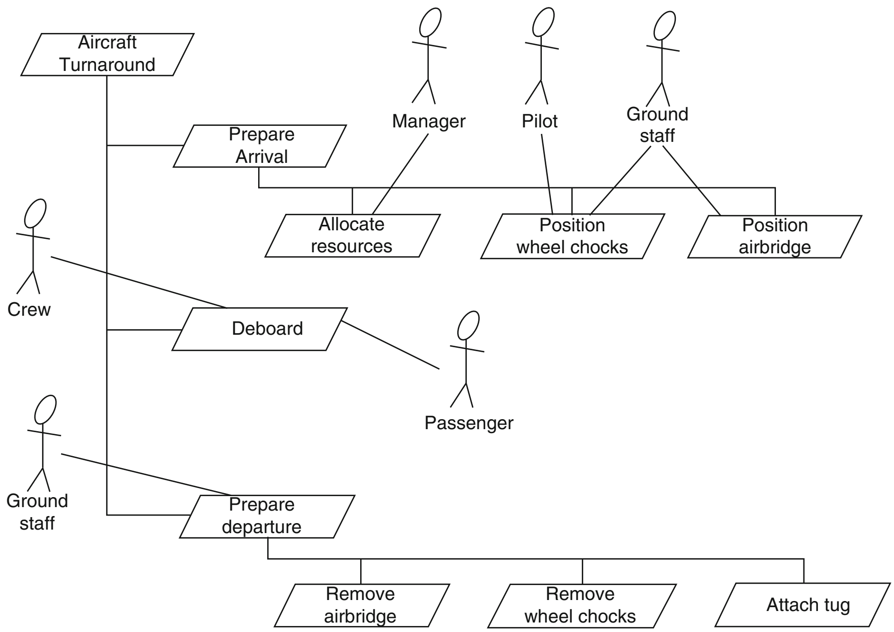


Fig. 2 A goal model decomposing the 'aircraft turnaround' goal

2. To add any relation between goals and sub-goals, `Goal has Goal`, defined in the ontology but not used in the model, and removing those not defined in the ontology.
3. To add to the model any relation between roles and goals, `Role responsibleFor Goal`, defined in the ontology but not used or for which there is no associated role in the model, and removing those not defined in the ontology.

Agent Models (e.g. Fig. 3) transform abstract constructs from analysis, e.g. roles, to design constructs, agent types, which are realised implementation [1, 15]. They describe the activities that each agent is involved in, along with their pre- and postconditions. The ontology validates that activities defined for each agent comply with the specification, that each agent plays the correct roles and participates in the correct activities using necessary environment entities to fulfil its goals. The validation consists of the following proposals:

1. To add to the model set any agents defined in the ontology but without corresponding models, and removing any agent models without corresponding agent defined in the ontology.
2. To add to every agent model, any missing activities associated with any of the roles (`Role participatesIn Activity`) played by agents (`Agent plays Role`), and removing any listed activities which are not associated to any of the roles played by the corresponding agent (as shown in the ontology).
3. To update the trigger or action fields to correct the pre- and postconditions of any activity in the ontology (`Activity precedes Activity` and `Activity follows Activity` respectively) whose pre- or postconditions do not match

Agent Type ID: CrewType
Name: Crew
Description: Play the role of the Crew.
References: R2 – Crew
Activity:

1. **Activity Name**: Crew embark aircraft.
   Functionality: Embarks an aircraft.
   Trigger: Receives embarking announcement from airline ground staff.
   Action: Informs passengers to embark.

2. **Activity Name**: Disembark crew.
   Functionality: Disembarks crew.
   Trigger: Passenger is perceived to has disembarked.
   Action: Announces to airline catering and cleaning staff that aircraft is ready for catering and cleaning.

3. **Activity Name**: Embark passenger.
   Functionality: Crew embarks passenger.
   Trigger: Receives announcement from airline ground staff that passengers are ready for embarking.
   Action: Announces to airline ground staff that embarking passengers is finished.

4. **Activity Name**: Disembark passenger.
   Functionality: Disembarks passengers.
   Trigger: Receives the announcement that airbridge is positioned.
   Action: Announces that passengers are disembarked.

5. **Activity Name**: Check aircraft.
   Functionality: Checks the conditions of an aircraft.
   Trigger: Crew has embarked aircraft.
   Action: Announces to airline ground staff that aircraft is ready for embarking passengers.

Environment considerations:
1. Airbridge.
2. Aircraft.
3. Flight plan.
4. Passenger Information.

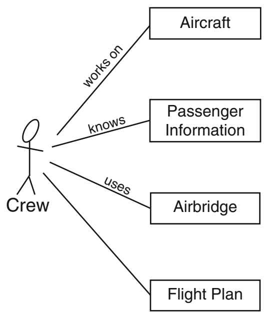


Fig. 3 Example of an agent model

any of the ones described by the fields trigger and action (any activity may have several pre- or postconditions). If the fields are incomplete, propose completion with the suitable activities as in the ontology.

4. To add to the environment list in every agent model, any missing environment entities used by any of the agent roles (`Role uses Environment`) or in any of the activities (`Activity needs Environment`) that the roles participate in (`Role participatesIn Activity`), and removing any listed environment entity not defined in the ontology as used by any of the agent roles or needed in any of the activities in which the agent participates.

4 Case Study: An Aircraft Turnaround Simulator

Aircraft turnaround refers to the process of preparing an arriving aircraft for departure. Typical operations that are involved are: Passengers disembark, luggage is unloaded, safety checks performed, then the activities for the new flight, loading food, luggage and embarking passengers are performed. The study arises from a Linkage project involving the third author. It is highly desirable to minimise the time that the aircraft remains in the airport, as longer stays mean higher costs for the airline. The MAS simulation is expected to identify how to optimise the process, completing a speedier turnaround with fewer resources (staff). Turnaround-related operations vary in duration and in how they are handed over within the sequence of tasks. There is scope for decentralisation and parallelisation. This makes the domain an excellent candidate for a MAS simulation.

We developed an ontology that models the problem as conceptualised by the client. We based on the documentation that the client provided us with, as well as several interviews with them. For the next step, we augmented the domain ontology annotating certain classes with concepts related to the MAS domain (see Sect. 3). Figure 4 shows an excerpt of the ontology and its augmented version.

Figure 5 shows the current state of the software development process involved and models interaction within this case study. The process is in its second/third iterations. The evolution of the models is clear so far: some models have already reached their final versions while others are expected to do so at the end of the third iteration. The validation process is iterative: models are validated as soon as they are developed and are revisited as soon as amendment proposals are reported by the iterative validation activity. This process proved to be effective as models are interrelated and therefore starting their development using corrected versions of the ones they are based on saves time as avoids rework. Iterations are undertaken until models converge and no further amendments are proposed by the validation activity. Due to lack of space, we cannot go into details for all the recommendations made, so we present illustrative examples of the process.

The initial set of models under validation included: environment, goal, role, organisation, interaction and scenario models. During the second iteration an agent model was added to this set. They evolved as follows:

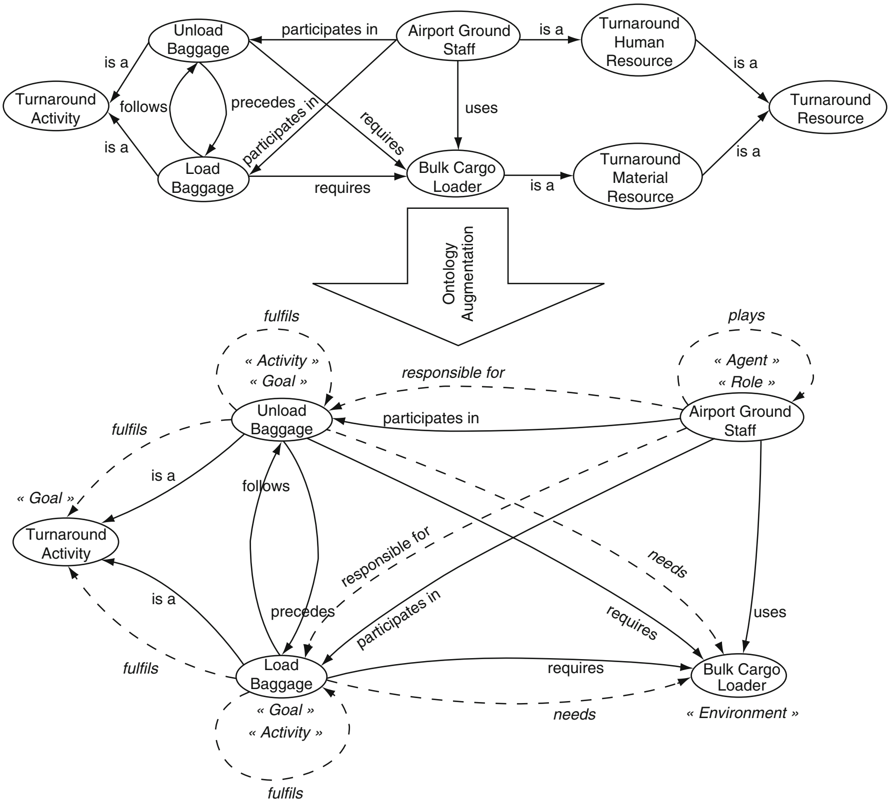


Fig. 4 Example of ontology augmentation: the above snap shot shows concepts and relations that are changed. Classes are annotated according to MAS concepts examined in Sect. 3

1. *Environment model*: In *Iteration 1*, this model lacked explicit relations between concepts and was not compliant with the ontology. Rework was proposed to improve it. In *Iteration 2*, it was changed thoroughly to be more faithful to the ontology. But still a few changes were proposed to align it with the ontology.
2. *Goal model*: In *Iteration 1*, this model was close to what was expected, only minor changes were proposed to improve it. In *Iteration 2*, it remained unchanged. Detected discrepancies had been suggested by the client beforehand. We consider this model validated.
3. *Role model*: In *Iteration 1,* this model was close to what was expected of it, so only minor changes were proposed to improve it. In *Iteration 2*, most of the proposals were accepted. Few remain pending further discussions with the client. We consider this model validated.
4. *Organisation model*: In *Iteration 1,* this model presented some inconsistencies regarding the hierarchical relation between roles. In *Iteration 2*, no improved version of this model has yet been developed.

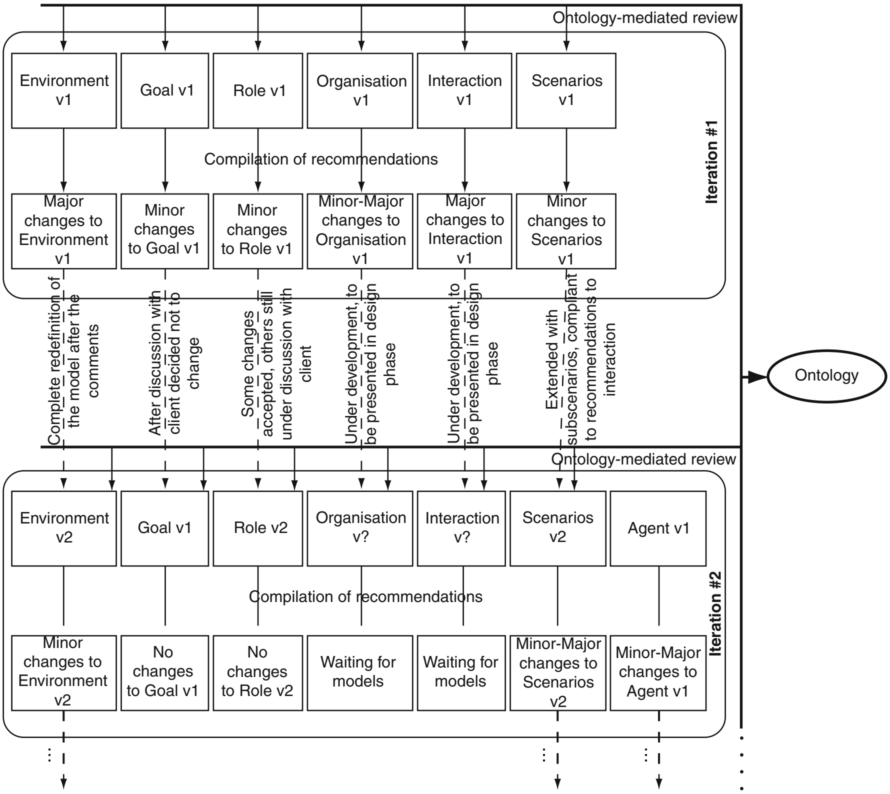


Fig. 5 Current state of development: the arrow head pointing towards the ontology entails that some aspect of the ontology were refined as result of validation process of MAS models itself

5. *Interaction model*: In *Iteration 1*, this model presented severe discrepancies with the ontology. Some interactions were not complete and others were not correctly planned (sequential or parallel). Most of the interactions were coordinated by the role Manager. Some of the potentiality of multiple processing by a MAS to be wasted. Agents could not interact autonomously with each other to achieve their goals. Using the ontology, we identified interactions that needed the mediation of the manager. In *Iteration 2,* this model is under development.
6. *Scenario model*: In *Iteration 1*, no recommendations were made to the scenario model, as a very basic version was provided. It included no sub-scenarios, only the main process was sketched. We suspected that upon its extension we would be able to propose amends (as it eventually happened). In *Iteration 2*, an extended version of the scenario was developed which included sub-scenarios detailing the turnaround process. These sub-scenarios reflected the suggestions made about the role of the manager in the interaction models. The scenario still has some minor flaws but generally speaking is correct. This proves the importance of concurrent work between validation and development activities to avoid rework.

7. *Agent model*: In *Iteration 1*, the agent models were not provided. In *Iteration 2*, this model was included for the first time. Some significant proposals are made. In particular some relations inter-models have to be improved and a few activities need changes regarding their triggers.

A third validation iteration is initiated, but yet to be completed. The number of proposals produced by the validation process has largely converged. The evolution of the models have produced high quality MAS models, and smoother interactions with the client indicating higher degree of satisfaction of client and developers.

5 Conclusion and Future Work

We apply ontologies to improve the quality of software models. Unlike other proposals we take into account the domain as specified by the client's requirements, filling any communication gap between clients and developers. Models are validated as soon as they are available, fixing errors as they arise and avoiding compounding and propagating errors to later phases of the development. To integrate our validation add-on seamlessly into the development process, we use an iterative, incremental and concurrent development process. The process iterates over intermediate versions of the model to achieve high quality. It is incremental in nature, not all the models are considered for each iteration. It is concurrent as development overlaps validation activities. This process can incur additional development cost and requires a cost justification. It is particularly appealing in critical software application where errors can be very costly and disastrous. This cost overhead may also be justified in the following scenarios:

- In developments of inexperienced modellers to guide them and avoid errors.
- In MAS developments of experienced modellers in any other technology, as agents have many particularities which cannot be found in other paradigms.
- In projects where the domain is complex or unknown, for experienced and inexperienced modellers alike.
- In software product line developments, where models have to be error free, as they will be reused in multiple developments.
- In projects dealing with the same domain, to enable reuse of the domain knowledge generated (i.e. the ontology).

That said, the cost of the validation can be greatly reduced by more effective reuse of existing ontologies. With advent of the Semantic Web, more ontologies are made available. More importantly, there is a great scope for generating the amendment proposals automatically. Indeed, we are now studying this possibility with the expectation to develop a tool that can significantly alleviate the burden of the details of the ontology-mediated validation process. In the future, we also intend to apply the ontology-mediated software model validation process to further cases studies to fine-tune it and to test our forthcoming tool.

Acknowledgements This work is supported by the Australian Research Council. The authors wish to thank developer Bin Lu for providing software models to support the case study.

References

1. Ashmalla A, Beydoun G, Low G (2009) Agent oriented approach for call management. In: International conference on information systems development, Nanchang, China. Springer, Heidelberg
2. Benevides AB, Guizzardi G (2009) A model-based tool for conceptual modeling and domain ontology engineering in OntoUML. In: International conference on enterprise information systems, Milan, Italy. Springer, Heidelberg
3. Benevides AB, Guizzardi G, Braga BFB, Almeida JPA (2009) Assessing modal aspects of OntoUML conceptual models in alloy. In: International workshop on evolving theories of conceptual modeling, at the international conference on conceptual modeling, Gramado, Brazil. Springer, Heidelberg
4. Beydoun G, Hoffmann A, Fernández-Breis JT, Martínez-Béjar R, Valencia-García R, Aurum A (2005) Cooperative modeling evaluated. Int J Coop Inf Syst 14(1):45–71
5. Beydoun G, Low G, Henderson-Sellers B, Mouratidis H, Gómez-Sanz JJ, Pavón J, Gonzalez-Pérez C (2009) FAML: a generic metamodel for MAS development. IEEE Trans Softw Eng 99(1):841–863
6. Brandão AAF, Silva VTD, Lucena CJPD (2007) Observed-MAS: an ontology-based method for analyzing multi-agent systems design models. In: Agent-oriented software engineering VII. Springer, Berlin, pp 122–139
7. Girardi R, Leite A (2008) A knowledge-based tool for multi-agent domain engineering. Knowl Based Syst 21(7):604–611
8. Guarino N (1998) Formal ontology and information systems. In: Formal ontology in information systems, Trento, Italy. IOS Press, Amsterdam
9. Hristozova M, Sterling L (2002) An eXtreme method for developing lightweight ontologies. In: Workshop on ontologies in agent systems, at the international join conference on autonomous agents and multi-agent systems, Bologna, Italy. ACM, New York
10. Juan T, Pearce A, Sterling L (2002) ROADMAP: extending the gaia methodology for complex open systems in international joint conference on autonomous agents and multi-agent systems, Bologna, Italy. ACM, New York
11. Lister K, Hristozova M, Sterling L (2005) Reconciling implicit and evolving ontologies for semantic interoperability. In: Tamma V, Cranefield S, Finin T, Willmott S (eds) Ontologies for agents: theory and experiences. Birkhäuser, Basel, pp 121–144
12. Nyulas C, O'Connor M, Tu S, Buckeridge D, Okhmatovskaia A, Munsen M (2008) An ontology-driven framework for deploying JADE agent systems. In: International conference on web intelligence and intelligent agent technology, Sydney, Australia. IEEE Computer Society, Washington
13. Okouya D, Penserini L, Saudrais S, Staikopoulos A, Dignum V, Clarke S (2008) Designing MAS organisation through an integrated MDA/Ontology approach. In: International workshop on transforming and weaving ontologies in model driven engineering, Tolouse, France. CEUR-WS.org
14. Shanks G, Tansley E, Weber R (2003) Using ontology to validate conceptual models. Commun ACM 46(10):85–89
15. Sterling L, Taveter K (2009) The art of agent-oriented modeling. In: Intelligent robotics and autonomous agents. MIT Press, Cambridge
16. Tran N, Low G (2005) Comparison of ten agent-oriented methodologies. In: Henderson-Sellers B, Giorgini P (eds) Agent-oriented methodologies. Idea Group, Hershey

17. Tran N, Beydoun G, Low G (2006) Design of a peer-to-peer information sharing MAS using MOBMAS (ontology-centric agent oriented methodology). In: International conference on information systems development, Budapest, Hungary. Springer, Heidelberg
18. Westland JC (2002) The cost of errors in software development: evidence from industry. J Syst Softw 62(1):1–9

Collaborative Media Content Management

András Ferenc Kovács, Gábor Magyar and Gábor Szűcs

Abstract The paper presents the design and implementation of an original integrated media content management system. In this very fast developing area where numerous changes occur even in a year's time it is really hard to foresee future trends and successful directions. The primary goal was to develop an example of a Collaborative Media Content Management System where different media types and other online contents can simply be stored integrated in one system and can easily be managed and shared by users and communities as well. Right at the beginning, we were aware of the fact that the project would last one year. Therefore, it was not enough to include the then newest features in the plan: it was also necessary to predict future trends and requirements, so that by the time the project is finished, it would still be innovative. The functional design of the application was based on the observation of new internet user generations who are heavily involved in real-time messaging and community portals. Based on our innovative ideas the system has been finished and besides our successful project many similar solutions have been appeared all over the world. The paper presents the most important features of our approach and the outcome: a Collaborative Media Content Management system, called simply Media-store, and emphasizes the innovative services of the system (visual tagging, mediashow, mediablog, common media arrangement).

A. F. Kovács (✉) · G. Magyar · G. Szűcs
Department of Telecommunications and Media Informatics,
Budapest University of Technology and Economics, Budapest, Hungary
e-mail: kovacsaf@gmail.com

G. Magyar
e-mail: magyar@tmit.bme.hu

G. Szűcs
e-mail: szucs@tmit.bme.hu

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_37,

1 Introduction

More and more websites offer some media content management solutions. Young generations are on the way to use media objects (photographs first) as a “language”, a kind of communication form. The phenomenon sometimes called as the emergence of “visual literacy” [1].

There was an opportunity for us—in a real-life company project—to design and implement a complex application in what a community of individual users can manage multimedia content. The output was—as we call it—a *community-class media content management* system.

The success of a portal where user generated content should be managed is largely dependent on the support from a content management application. A portal, where members are interested not only in publishing their specific individual content but in creating common media topics and expressing something common by grouping, tagging and blogging objects together—such a portal should provide community content management solution [2]. Innovative services of our system includes a special metadata management subsystem for visual tagging [3], “mediashow” for easy creation and publication of real multimedia experience, mediablog for visual (or “sensual”) blogging, rather than only text commenting. Furthermore common media arrangement was introduced by what a community can continuously find and express it’s interest and values [4].

2 Functional Design Considerations

There is no room and need to describe the full functional planning. We would like to emphasize only features necessary to understand the main characteristic of the system, especially those what implement collaborative functions [5].

The Media-store is planned for three hierarchical person groups: visitors, users and administrators as can be seen in Fig. 1. Visitors are the persons, who use the public part of the system without registration. Users are the persons, who successfully enter the system with name and password after the registration. Administrators have the highest level rights, they may analyze the statistics, moderate the remarks, delete the persons or groups. The functionalities of the Media-store can be divided into two parts: functionalities related to the contents and related to the operations [6]. The Fig. 1 shows the most important functionalities according to the different actor levels.

Visitors can visit to the public albums they may view the previews of the public media contents. If they have corresponding rights they can search among the contents, furthermore they may view the public blogs. The visitors can send a link about the most interesting public contents by e-mail. If they would like to use the other features of Media-store, they should register.

Users can upload and manage contents; they can view the shared private contents, may evaluate them and may write remarks about them. The users can moderate

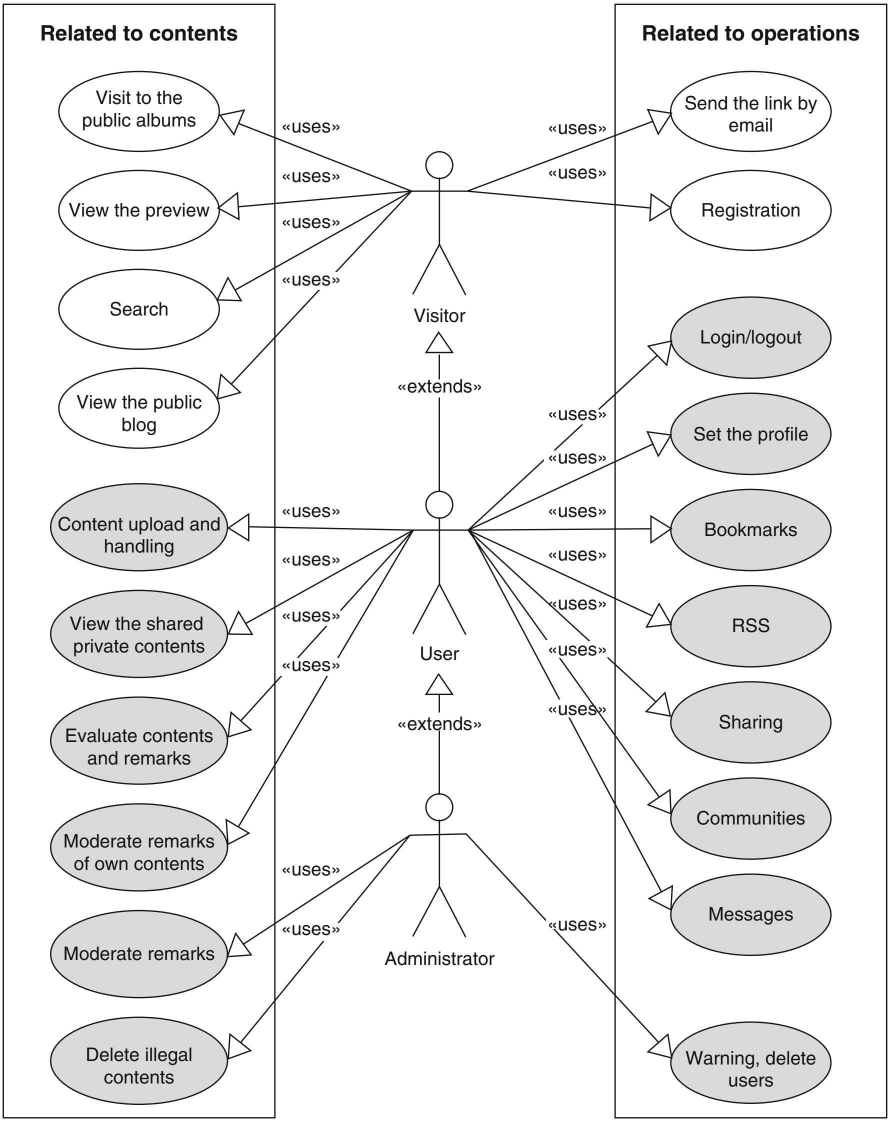


Fig. 1 Use case diagram showing actors and the most important functionalities

remarks of own contents. The users can login, logout, set own profile, create and delete bookmarks, use RSS (Really Simple Syndication). They can share own media contents, send and receive messages. The users can establish special groups, so called communities, where they can communicate with each other, can share the common contents and can work together (see later at the collaborative handling of the media contents).

The administrator can moderate remarks of contents as well, and they may delete illegal contents. Furthermore the administrator can send warnings to users and may delete the users.

The usage modes of the contents in the Media-store from users' viewpoint are the following:

- The users can *view/listen* the contents if they have corresponding rights.
- The users can *download* the contents if they have corresponding rights.
- The users can *copy* the contents if they have corresponding rights.
- The users can *create a link* to the contents if they have rights to view.

The Media-store was designed to store different types of media, like image, video, audio. The key for the well arranged and searchable content-store is the metadata system. For this reason the metadata of the different types of media was worked out thoroughly. A visual tag concept was introduced. This tag is able to present the names of the persons in different parts of an image. There is another comfortable function for the users: automatic completion of words, where in the filling process of metadata the previously used phrases are offered to users automatically by the system.

3 Logical Design and Implementation of the Media-Store

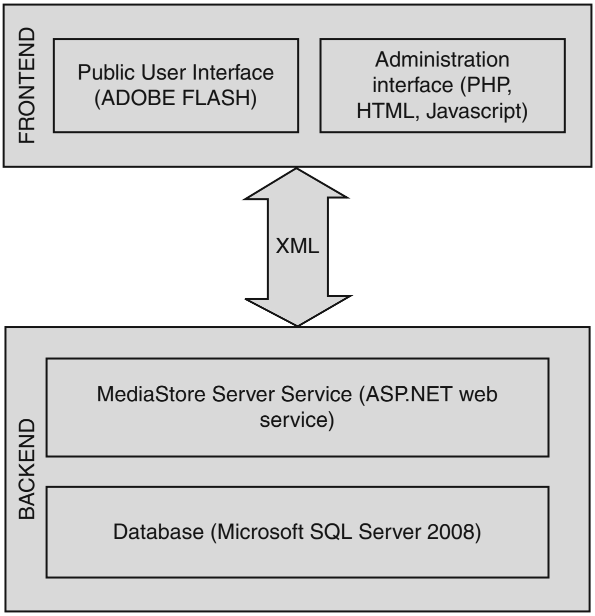
3.1 System Architecture

The whole Media-store integrates several individual subsystems and modules that are responsible for different tasks in the system. Every subsystem was implemented in an environment that is the most suitable for that particular task. The complete system structure which follows the standard 3-tier architecture is shown at Fig. 2 below.

The front-end of the system was divided into two different interfaces based on the target usage and their requirements. These are the public user interface and the administration interface. The public user interface used by the visitors and the public users was implemented in Adobe Flash environment to maximize compatibility, platform independency and user experience. The administration interface for the operators where the functionality, speed, robustness, stability, reliability and preciseness are more important than the design and experience was implemented using simple standard HTML, PHP and Javascript technologies.

The back-end of the system is based on a standard web service, the MediaStore Server Service which was implemented as an ASP.NET web service running on an IIS7 application server. Between back-end and the front-end standard XML based communication was designed which on the one hand gives perspicuous and industry standard interface for the back-end and on the other hand makes front-end development independent from the back-end and easier even on very different platforms like mobile phones, set-top-boxes or other special target devices in the future.

Fig. 2 System architecture



Except the binary data of the media files all the data are stored in SQL database. Direct data manipulations were written using stored procedures executed by the database server and called by the service. This speeds up complex queries and separates data access layer from the logic layer in the web service. Media files are stored directly in the file system and identified by unique file names comes from the id of the media object in the database [7].

3.2 *Media Types*

The system was designed to store three different types of media: images, videos and audio files. All the commonly used and spread file formats for the different media types are supported by the system. The complete list is shown in Table 1.

Maximum size of individual files that can be uploaded is 2 MB for images and audio files. Uploaded videos are converted to flash video format. The size of these flash video files must not exceed 10 MB. These limits can be set on the administration surface, values mentioned are the defaults.

Table 1 Supported file formats

Images	bmp, jpg, gif, png, tif
Audio files	wav, mp3, wma, amr
Videos	All input formats converted to flv during the upload process

The system differentiates several categories between similar media types to help users searching contents for different use. An image for example can be a photo, a drawing or wallpaper; a video can be a film, home video or animation; an audio file can be ringing tone, music or voice recording etc. These categories are built in the system and one has to be chosen by users during the upload procedure for every media files.

3.3 Images with Visual Tags

One of the innovative features of the metadata system is visual tagging. This is an assignment of a custom caption and a selected area in an image. This caption can be the name of the person marked in the picture or just a word or small text describing the content. The person can be chosen from the registered users as well. In this last case we say that the user is tagged in that image.

Every public picture can be tagged by all registered users, but any tags can be removed by owners. For the reason of personality rights user-tags must be accepted by the tagged user before they are published and even later they can be removed from any public pictures also. To help users and speed up adding visual tags automatic completion was implemented in the process. It suggests previously added phrases and names of registered users when typing.

These tags are stored individually as two Descartes coordinates and the caption linked to the content. The caption can be a user ID or custom text. Sound, picture and motion picture tags can be introduced in a later phase of the development.

3.4 Mediashow

Mediashow is an innovative new feature in the system. It is similar like slideshows made from pictures, but here all the media types managed by the store can be mixed in one united mediashow. Pictures and videos are displayed or played in album order, one after the other.

For pictures display time, dissolve effect and time between two pictures, the length of the effect animation, or music played during the mediashow can be set. Video and audio files can be set to be played automatically or by user interaction. For audio files a visualization procedure can be chosen to show when playing. For every type of media captions can be added to display on the top, bottom or the center of the screen. Font size can be set also.

Practically in the system mediashow was implemented as a view of an album of selected media files with the possible settings mentioned before. These mediashows can be shared by owners to other users in the system or can be made available by a public unique link pointing to the selected show. For security reasons the validity of this link can be set to limit the access to the mediashow in time or view count.

The time can be a day, 3 days, a week, a month, a year or unlimited, view count can be a number from 1 to unlimited. Using these features the well selected and formed memories can be shared even with non registered visitors [8].

3.5 Blogs

In the system a slightly different type of the common blog service called mediablog was implemented. This means that the primary focus was put on media contents instead of text entries here. The main purpose of this feature is to establish the base of visual or fancier sensual blogging. Still simple text can be added but any public or own media contents can be easily included from the store to increase the emotional or imaginable value of the entry.

The subsystem itself was implemented like a standard blog engine. Individual blog templates can be chosen or built up, and unlimited amount of entries can be added by the users. As a special feature users can register their mobile phone numbers and then they can send SMS or MMS messages to the blog system. These messages are identified by the registered phone number and automatically assigned to a user profile. Messages can be set to get automatically published as a blog entry or just stored in the system for later manual publication. An email address can be registered as well for this feature. E-mails sent from this address to the system can also be set to be published or stored.

As a widely spread feature comments can be added to any public blog entries where formerly this option was set by the owner in the blog's profile. Naturally unwanted comments can be moderated by the blog owners as well if necessary.

As formerly mentioned communities can be founded by a group of users in the system which can also have mediablogs. In this case authoring rights can be assigned to several members of the community while others are just readers in the blog. Several mobile phone numbers and email addresses can be registered as automatic publication sources also. Rights can be controlled by the creator owner of the community or by users with the same role. If some users would like to use their mobile phones or email addresses in different blogs as a source the target blog can be chosen by adding its name as the first word in the message after a hash mark [9].

4 Collaborative Managing of Media Contents

As the main goal of the development was to create a "full-blooded" Collaborative Media Content Management system a flexible community administration module was designed and implemented. Users can create, manage, and delete communities in the Media-store (what is quite typical feature nowadays in Web 2.0, but should be mentioned: at the beginning of our project this was novel) [10].

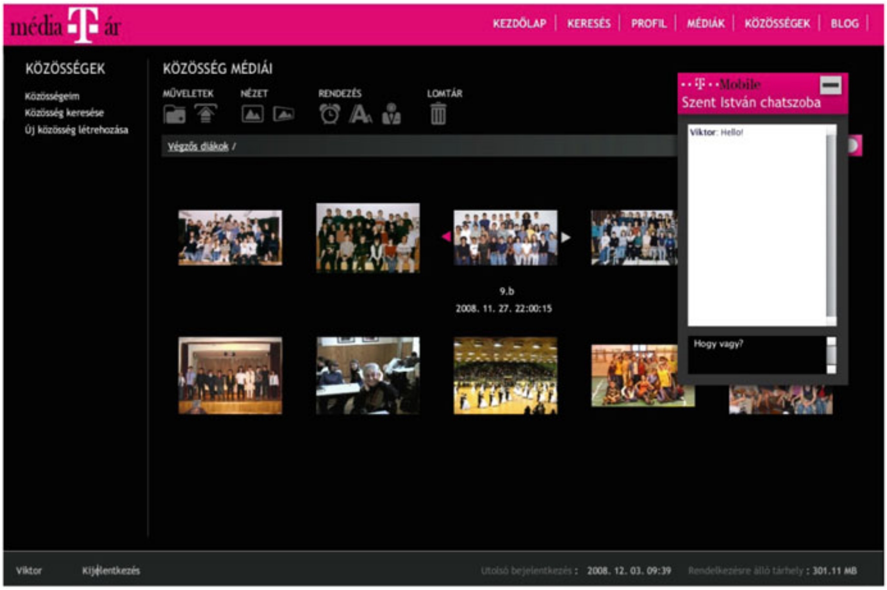


Fig. 3 Screenshot from the implemented system

After a successful login procedure the users can use the chat rooms (this popup window appears at the beginning of the navigation of the community pages as can be seen in Fig. 2), and can talk to each other within a community. The connection to the chat room is established automatically when users navigating to the community pages and disconnected when leaving. In Fig. 3 the thumbnails of the common pictures, chat room, the main menu and the side menu, furthermore the status bar at the bottom of the screen can be seen (the screenshot contains Hungarian labels as the product was developed to the local market).

The most innovative function in the Media-store is the common arrangement, which gives community members the ability to manage media contents of the community not only online but in real time collaboration as well. By this implemented function the pictures can be rearranged in the albums by a member of the community while all the other members of the same community will see the modified layout of the thumbnails in real time. The common arrangement is supported by the chat room, where the instructions about the layout modification can be commonly discussed.

5 Summary

A collaborative media content management system was fully designed and implemented in the project which was initiated by a multinational telco, carried out by a research team in the university.

In information technology the media content management is a very fast developing area where the future requirements of the market can be hard predicted so the functional design of our solution had to be based on trends and probable future needs of users. Therefore the functional plan has been built on the observation of habits of the new internet user generations who are heavily involved in real-time messaging and community portals [11]. Those youngsters are becoming the people of the screen rather than of the paper sheets and/or textual messaging.

We have defined the community portal as a content-infrastructure what should be able to aggregate and deliver the content in a highly personalized, secured, in-context manner, and should enable users to collaborate or make the content delivered actionable. Our collaborative media content management system provides features not only to add, edit, and delete the content but to use content objects as visually tagged building blocks of a community. The flexible framework enables communities in different size to easily manage media assets and the profiles of the users. The conclusion of our work is that the innovative ideas are not enough to the success; we have integrated them into a unified content management system with modern, improving technique, the flash. The flash gives so visual experience to the users (not only at videos, like in YouTube, but in portals too), which will be totally usual in 1–2 years, furthermore this helps with content handling as well.

The final result is an efficient way of handling the content throughout its lifecycle from the time of creation to disposal. We propose to call that approach as *community-class media content management*. That kind of media content management empowers users in the boundless virtual communities to participate in content creation and provides funny collaboration patterns. The model that maps to the structure of the virtual communities is scalable.

Based on our innovative services of the system (visual tagging, mediashow, mediablog, common media arrangement) the long work has been finished and besides our successful project many similar solutions have been appeared all over the world [12]. E.g. in Hungary the largest social network system—“iwiw”, operated by the telco company what was our partner in this development—has been applied the visual tagging based on our joint results in Media-store.

End users of Media-store are supported by an easy-to-use interface to submit and tag content to their site, detect and influence changes. Real-time community contribution for content evolution matches to the “online lifestyle” of young generations.

References

1. Institute for the Future (2009) The future of video map of opportunities. http://www.iftf.org/FutureofVideoMap. Accessed 14 Dec 2009
2. Olsson T (2009) Understanding collective content: purposes, characteristics and collaborative practices. In: Proceedings of the 4th international conference on communities and technologies, University Park, PA, USA. ISBN: 978-1-60558-713-4

3. Halpin H, Robu V, Shepherd H (2007) The complex dynamics of collaborative tagging. In: Proceedings of the 16th international conference on the world wide web (WWW'07), ACM Press, Banff, Canada
4. Nakano R (2002) Web content management: a collaborative approach. Addison-Wesley Professional, Boston. ISBN-10: 0201657821
5. Latham L (2007) Key issues for web content management, gartner analysis 15 March 2007. Stamford
6. Boiko B (2002) Content management bible. Wiley, New York. ISBN-13: 978-07645 48628
7. Davidson L (2008) Pro SQL server 2008 relational database design and implementation. Apress, New York. ISBN: 978-1-4302-0866-2
8. Bhana I, Johnson D (2005) A peer-to-peer approach to content. Dissemination and search in collaborative networks. Lecture notes in computer science. Springer, Berlin. ISSN: 0302-9743 (Print) 1611-3349 (Online) Volume 3516/2005. ISBN: 978-3-540-26044-8
9. Vander Wal T (2007) Folksonomy coinage and definition. http://vanderwal.net/folksonomy.html. Accessed 12 May 2010
10. Hummel J, Lechner U (2001) The community model of content management: a case study of the music industry. University of St. Gallen, Switzerland. doi: 10.1080/14241270109389940
11. Pereira CS, Soares L (2007) Improving the quality of collaboration requirements for information management through social networks analysis. Int J Inf Manag 27(2):86–103
12. TopTenREVIEWS (2010) Video share websites review. http://video-share-review.toptenreviews.com. Accessed 10 May 2010

Modification of Workflow Management System Focused on a Human

Vojtech Mates

Abstract This paper discusses a problem of managing resources by utilizing the Workflow Management System taking into account the properties of WfMS participant. The paper shows a problem using architecture according to Workflow Management Coalition—no distinction is made between participants of Workflow Management System with the same role. The possible solution lies in the modification of the process definition. The paper shows how to use the Workflow Management System for setting up dynamic properties of profiles of WfMS participants which significantly supports the solution for more precise managing resources in Workflow Management System. This modification of the Workflow Management System also enables other ways for the analysis of the business process. The solution can also improve assigning participants of Workflow Management System to task.

1 Introduction

At the beginning, it should be mentioned why it can be useful to use the Workflow Management System (WfMS), for description of workflow see [1]. An important motivation for using WfMS is supporting process-oriented management, where processes are more important than the hierarchy in organization. This attitude provides better utilization of resources in organization.

The Workflow Management System is a tool used for automation of business processes. It can manage a mutual cooperation of the WfMS participants (including human and software interaction) in order to perform a business process.

The main purpose of WfMS lies in coordination of resources according to the process definition which is usually created by the company management. Making changes in a business process is easier, because the process definition is not hardcoded into the system, thus the company is more adaptable to changing conditions.

V. Mates (✉)
Faculty of Information Technology, Brno University of Technology, Brno, Czech Republic
e-mail: imates@fit.vutbr.cz

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_38,

WfMS also provides a good background for business process re-engineering, because it also provides valuable audit data (in addition to the management and monitoring of the processes), which can later be used for analysis. Extracted information from audit data can be also used for better managing as it will be showed later.

The goal of this paper is to show some changes in WfMS architecture which can improve the system as the whole in particular areas. There will be description of how to provide dynamic updating of certain properties of participants of WfMS using WfMS.

The solution lies especially in setting up the dynamic profiles of WfMS participants and changes in the process definition.

2 Architecture of Workflow Management System

The architecture of WfMS will be described very briefly according to the reference model created by Workflow Management Coalition (Fig. 1).

According to [7] Workflow Management System should consist of:

- Process Definition Tool,
- WFM Engine,
- Worklist Handler,
- User Interface.

The creation of the process definition (for example BPMN, see [2] and [15]) is the first part of the process deployment in the Workflow Management System. This is usually done by the Process Definition Tool, see [11]. A process described by the WFDL (Workflow Definition Language) is an output of this tool. XPDL (XML Process Definition Language) and BPEL (Business Process Execution Language) are usually used as WFDL. The description and the comparison of languages for the description of the process can be found in [12] for modeling and transformation of workflow see [6].

The process definition should include information about tasks including information such as who can perform the task (mostly the role-based approach) and information about the routing between the tasks. The business process execution logic can be described as routing mentioned later more in detail.

The process definition is transported to the WfM Engine, where the process definition can be launched as the process instance. Instances (particular processes) originate from the process definition. Worklists are created by running particular instances according to process definitions with specific data. For instance, during the execution, applications or humans can update a workflow's relevant data.

Worklist Handlers manage an assignment of the task to resources e.g. human or software (mostly web services) after launching the process instance.

The WfMS participant is an object in WfMS which participates in the process instance (usually human or web service). The user communicates with the system using several user interfaces like email, application interfaces, etc.

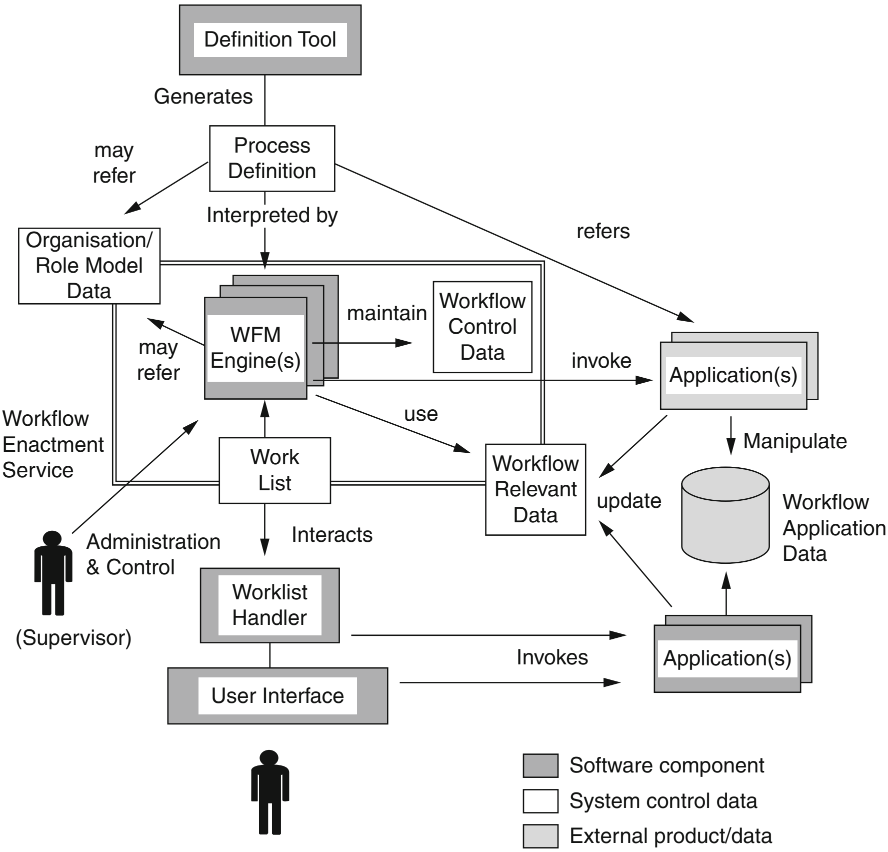


Fig. 1 Architecture of Workflow Management System, see [7]

All information about the performance of particular processes can be audited for monitoring and analysis purposes.

More information about WfMS architecture can be found in [7], [8] and [17]. More information about BPM, which uses WfMS as the tool can be found in [4].

3 Modeling Workflow Using Petri Nets

Petri Nets formalism is widely used to model dynamic aspects in systems. Petri Nets consist of places, transitions and arcs. Places represent conditions in the workflow. Transitions represent the performing task. Oriented arcs connect places with transitions. We can simulate performing a process under conditions by moving tokens from places to places within Petri Net. More information about Petri Nets can be found in [5].

We use specific patterns in Petri Nets to model business process such as: AND split, AND-join, OR-split, OR-join, more patterns in [16]. According to [13], there are four types of routing:

- sequential,
- parallel,
- selective,
- iteration.

For sequential routing, it is necessary to perform one task after another, because there are dependencies between them. All tasks run concurrently in parallel routing, construction consists of AND-split and AND-join. In selective routing, just one branch is performed, construction uses OR-split and OR-join. In iteration routing, certain parts of the Petri Net are repeated until the appropriate condition is passed.

Hi-level Petri Nets Formalism also supports:

- color,
- hierarchy,
- time (time stamps).

Color extension enables token holding several attributes of certain type. Hierarchy enables building process from sub processes. Time extension can be used for time related conditions, more in detail in [5].

More information about modeling business processes can be found in [13] and [9], about dynamic change modeled in Petri Nets in [3], about detection deadlock and traps in [14].

4 Basic Attitudes to Assignment in WfMS

There are two basic attitudes to assigning the WfMS participant to task: manual assignment and automatic assignment based on role mentioned in the process definition.

There is a person who is responsible for the assignment of the item from the worklist to the WfMS participant in manual assignment. Manual assignment has the main advantage in taking into account WfMS participant properties. The person is able to distinguish between particular persons. But there are also some disadvantages:

- The person who makes an assignment can be overloaded (queues).
- The person has to know properties of all WfMS participants very well in order to utilize the main advantage.
- The manual assignment usage is more expensive and slower than automatic assignment.

Another way is to automate the assignment items from the worklist to WfMS participants. There is the information about which group of the WfMS participants can manage the task within the workflow (role-based approached) in a process definition. WfMS participant is chosen from a group which is capable to perform task just before performing the work item.

Automatic assignment according to [7] has advantage of saving cost. Performance of the system is also higher. On the other hand, we loose the ability to choose resources more individually. The improvement of automatic assignment is a decision based on more suitable parameters of WfMS participants.

5 Problem with Handling WfMS Participant in WfMS According to WfMC

There are usually several ways how to perform certain work, but Workflow Management System can not distinguish between WfMS participant from the same group and thus it deals with them just according to participation in certain groups. Therefore, the system forces the interaction with a beginner in the same way as with an expert. Why could it be a bad behaviour? Let's try to answer very simple question: How many checkpoints is necessary for making subprocess perfect for a beginner and an expert in a particular task at the same time?

It is necessary to keep some level of quality of the process in order to check if everything is going well in certain states of performing the task in order to reveal mistakes as soon as possible (at the worst case at the end of subprocess).

Let's suppose the process definition perfectly fits to the beginner. Now the process is perfectly set up for the beginner, but the process has just made less suitable for an expert at the same time. The expert will not be comfortable with the system and his productivity can even decrease as a result of it. The reason is too much interactivity with the system or loosing freedom in performing a subprocess. The expert naturally does not need such a care of WfMS as the beginner. The expert can keep low level of making mistakes during performing task.

Let's try setting up the process according to the expert (process will look probably much more simple, for example less checkpoints), but we have just made another and maybe a worse problem by decreasing checkpoints. The beginner has now too much freedom, thus mistakes made by him will be revealed much later. It can significantly decrease performance of the beginner, because he will find the mistake at the end of his job at worst case, which means that the part of the process has to be repeated. It also implies another allocation of resources using during performing part of the process, because mistakes revealed later means larger costs.

Another problem can be, for example, related to different nationalities. Every nationality has a different cultural background, therefore WfMS should be able to deal with these characteristics as well. The age and education of the human participant can be also important in certain cases etc.

6 Modification of the Behaviour of WfMS

In order to solve preceding problem we need to change some parts of the Workflow Management System. At first, we need to change the process definition in order to support the distinction between WfMS participants even if there are placed in the same group. Placing WfMS participants to group represents that a WfMS participant has a certain qualification for performing certain tasks.

Let's suppose an assignment of the participant to the subprocess has just been done and all his properties can be read and the right managing can be set according to his current set of properties (similar to a manual assignment).

We can enrich the process definition with the rules using certain properties of the WfMS participant. Thus, we can choose the right route in the subprocess according to their particular properties of the WfMS participant.

7 Managing Profile of WfMS Participant

In order to support the process definition with an extension containing rules based on properties of the WfMS participant, up-to-date profile (the set of properties typical for the particular WfMS participant) has to be managed. Let's separate to groups of properties:

- static properties,
- dynamic properties.

In case of static properties, it is possible to manage them manually (editing records manually); these properties do not have the specific relation to company, for example age, sex, education, nationality, etc.

In case of dynamic properties, it is very difficult to keep the right values up-to-date. Examples of dynamic properties are: experience with particular task, error rate, experience with group of task, etc. As we can see, dynamic properties describe the behaviour of the WfMS participant in an organization. It would be very helpful if it was possible to distinguish between a beginner and a more experienced user, thus dynamic properties is needed. But updating dynamic properties manually would be almost impossible without automatic support.

7.1 How to Get Dynamic Properties

The problem with updating of dynamic profiles can be solved using WfMS. Due to running WFM Engine the data about behaviour of the WfMS participant can be stored. It is needed for the review of history of the particular process. Thus information can be extracted about:

- the error rate,
- the experience with particular task,
- the experience in his profession.

The error rate can be found out by counting positive and negative result of the test in checkpoint. In case of positive or negative results, the particular counter is incremented. Of course, the result closer to present time has higher weight than result which was made more in the past.

The experience with the particular task can increment the specific counter for the participant and the task. It should be also taken into account how far data were created in the past. Data created in present are probably closer to reality. Experience with certain group of tasks is example of the monitored indicators of overall quality of the WfMS participant.

In order to manage dynamic properties, updating specific values is triggered by events which are created by WfMS. Events are happening during performing the instance of the particular process.

7.2 WfMS Participant Profile

The static profile of WfMS participant is a set of properties with declaration of static properties which are valuable for the company. The information can be updated by filling-in some forms by WfMS participants mostly at the beginning of a participation in the company and later manageable by office administrators or by themselves.

The dynamic profile is done by using WfMS. Events created by performing the instance of the process trigger updating of dynamic profiles. The structure holds collection of an item related to particular tasks. Every item holds set of monitored indicators (error rate, level of experience, etc.) for each task. It contains collection of items containing data to group of tasks (error rate, experience) as well (overall experience), more in [10].

8 Modification of the Process Definition

The motivation for the modification of the process definition is mainly due to better managing WfMS participant within WfMS. Process definition contains in general:

- information about the task in the process (task properties, role of resource)
- and routing (rules used to specify an order of the performing task sequential, parallel, selective, iterative).

In order to solve the problem mentioned above, extension of routing information is needed (especially guard functions and arc expression functions to include variables based on properties of resource). Let's suppose all important properties are know including error rate, experience with certain task due to making dynamic profile which contains up-to-date error rate and experiences and the change of the definition of the business process is possible as well.

We can model this extension in Hi-level Petri Nets (commonly used as process modeling formalism):

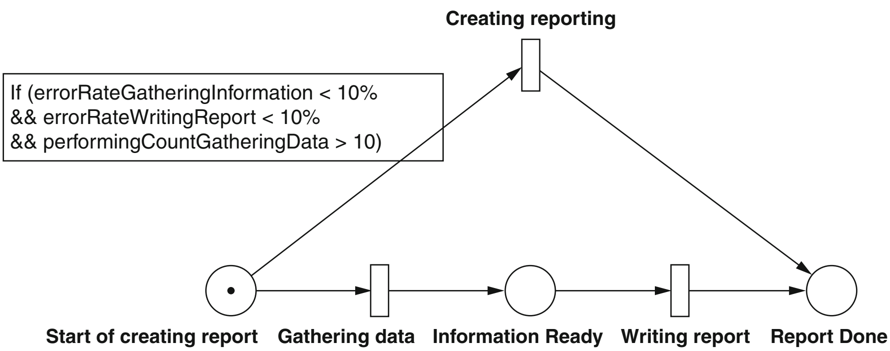


Fig. 2 Example of changed process definition

- We use current model process definition built from patterns, see [16].
- Token has to carry data necessary for performing task and routing information. It is necessary to add information containing set of properties of WfMS participants assigned to the task to token in order to use it for the routing.
- It is also necessary to add guard functions, arc expression functions which will use WfM properties as the input.
- New subnets of Petri Net can be built according to the preceding rules.

As we can see, we are still using Hi-level Petri Nets even if the extension enabling interaction with the WfMS participant based on his personal properties was added which means that the same analysis method can be used as we could use before the extension, but we gain the process definition which can describe different behavior depending on WfMS participant properties. Thus, new rules, which can work with properties of WfMS participant, can be added.

8.1 Example of Modified Process Definition

In Fig. 2, the sub process of creating report is presented. The expert is able to make a report as one single step. He can combine gathering information and writing. He can recognize correct data. But the less experienced user will be checked at the first checkpoint if he has correct and relevant data before he will continue with writing the report. Because the user has less experience the data has to be checked before he starts with writing. In case of failure in the first step, performing the second one is not necessary and the subprocess can be immediately restarted.

By repeating creating of the report, the user gets knowledge and constantly improves his performance. Getting experience simulates updating his profile after every step he made. There is a test at the end of each step. In case of a success or a failure, particular counter is increased. In case of successful performing, particular counter is increased.

In case a certain condition is met, he becomes a more experienced user and he can skip one checkpoint. In our case the goal is to get the error rate in gathering information under 10.

In case of performing instance of process, the WfM engine checks if the condition based on the profile of the assigned WfMS participant is met in order to choose the right route.

Due to the extension WfM Engine can interact with an expert in a different way than with a beginner even if they are in the same group that qualifies them for the performing certain tasks. For example, the more experienced can have the different route with different number of checkpoints.

Thus WfMS changes its interaction to particular user dynamically according to his profiles. For example if the user managed to reach some number of performing of the subprocess and has an acceptable error rate at the same time, then WfMS would deal with him in a different way than before.

9 Adaptation of WfM Engine

From the implementation point of view meta data of the process definition has to be changed and adaptation of WfM Engine for performing the modified process definition is also necessary. In case the WfM Engine was supporting Hi-level Petri Nets, no significant changes are necessary, because WfM Engine will just work with more parameters and with larger Petri Nets. Even if the Hi-level Petri Nets changed in order to support WfMS participant extension it would be described by Hi-level Petri Nets which could be interpreted in the same way as it was before.

10 Adaptation of Worklist Händler

By creating dynamic profiles of WfMS participant, we made benefit, because we gained an option to distinguish between WfMS participants from the same group (for example an expert vs. a beginner). We can also improve the assignment algorithm using the extended model. According to setting priority of business process (cost, time, quality), we can make automatic assignment based on properties of the WfMS participant.

We can set up our priorities in the automatic assignment. We can prefer an expert in the case we need to decrease the error rate. We can prefer a cheaper resource in the case we are not in hurry, thus we can try to save costs. It can be a good strategy to assign the beginner to task in the case we can afford it, because we also need to make regular updating skills of the WfMS participant and save experts for more difficult situation.

WfMS participants impact performing of process and process impacts the WfMS participant (performing the process also updates skills). For example, if we are expecting waiting in the synchronization point, we will be able to use less experienced user with small impact of overall performing of the process.

Parameters of the process can be static or dynamic. Static parameters can be set up at the beginnings as the input to the assignment algorithm. Let's suppose these basic parameters:

- cost
- speed
- quality.

We can detect divergence currently running instance of a process from common performing of the process, because we know history of the preceding performing of the particular process. Thus we can reconfigure parameters of the instance of the process dynamically based on the current performing of the instance. Changing parameters impacts the assigning resources to tasks.

11 Modification of Audit Data Created by WfMS

The using extended model focused on properties on the WfMS participant is very useful for business process analysis as well, because we can record current state of WfMS participant during running instance of process. The profile at the moment of the assignment represents the current of the WfMS participant (set of properties).

Due to triggering mechanism of WfMS the profile holds up-to-date information about the current state of WfMS participant. Because WfMS knows up-to-date profile of all WfMS participants, it can also record the exact state of the WfMS participant due to auditing data at the moment of recording event. Thus we gain option to make analysis which work with properties of the WfMS participants.

For example, due to changing audit dates, we have the possibility to answer the following questions:

- How many times has the WfMS participant had to re-do a particular task in oder to get the error rate less than five percent?
- Which properties of the WfMS Participant have the best results in particular task?
- Which of the set of properties has the largest impact on the best performing?

Why do we need to analyze properties related to particular tasks? There are several reasons:

- We can manage changing properties of WfMS participants in order to adapt to changing conditions, because we can educate, manage participation in certain kinds of processes, etc.
- We can find out, if someone has a worse results in certain groups of task and if someone has better results in other groups of tasks related to others. As a result of analysis, we can change membership in qualification groups.

- We can change the extended process definition based on the analysis. For example, we can change a value in conditions important for dealing with WfMS participants as with an expert.

There are more reasons why it is helpful to analyze the properties of WfMS participants. The need for that kind of analysis is obvious, because the results of processes are significantly impacted by WfMS participants. The problem of having less knowledge about particular WfMS participants is even worse in process-oriented management, because we try to reduce the hierarchy as much as possible which means that managers are losing feedback to workers.

12 Conclusion

The goal of this article was to describe an idea of extension of the Workflow Management System architecture in order to make better management of WfMS participants. The extension improves modeling abilities at describing business processes more precisely and thus WfMS deals with the WfMS participant more individually.

The presented solution lies mainly in changing the process definition and managing dynamic profiles of WfMS participants. It was also presented how to use current WfMS to manage dynamic profiles which are important for the presented extension. Using profiles also enables changing the algorithm for the assignment which supports the dynamic reconfiguration of priorities of the process. The extended model gives us also the possibility of analysis which can answer several important questions for a future evolution of WfMS participants in a company. The solution was presented as the modification of the architecture presented by the Workflow Management Coalition. The changes in the process definition was very briefly shown on Petri Nets.

References

1. Cichocki A et al (1998) Workflow and process automation: concepts and technology. Kluwer, Dordrecht, p 117
2. Dijkman RM, Dumas M, Ouyang C (2007) *Formal* semantics and automated analysis of BPMN process models. http://eprints.qut.edu.au/7115/1/7115.pdf. Accessed 12 Oct 2009
3. Ellis C, Keddara K, Rozenberg G (1995) Dynamic change within workflow systems. http://portal.acm.org/citation.cfm?id=224021. Accessed 20 Dec 2009
4. Fischer L (2008) BPM & workflow handbook—spotlight on human-centric BPM, 1st edn. Future Strategies, Inc, Berlin, p 330. ISBN-13: 978-0977752768
5. Girault C, Valk R (2002) Petri Nets for system enginnering, 1st edn. Springer, Berlin, p 607. ISBN-13: 978-354041217
6. Gruber W (2005) Modeling and transformation of workflows with temporal constraints (Disdbis) Dissertationen Zu Datenbanken Und Informationssystemen). IOS Press, Inc, Berlin, p 168. ISBN-13: 978-1586034160

7. Hollingsworth D (1995) Workflow management coalition the workflow reference model, United Kingdom. http://www.wfmc.org. Accessed 20 Jan 2010
8. Hollingsworth D (2003) WfMC the Workflow Reference Model 10 Years On, United Kingdom 2003. http://www.wfnc.org. Accessed 15 Feb 2010
9. Marlon D, Van Der Aalst W, Hoftstede A (2005) Process-aware information systems. Wiley, Hoboken, p 432. ISBN: 978-0-471-66306-5
10. Mates V (2010) Using Workflow Management System pro Analysis Based on Properties of Resources. In: Petr Šaloun (Ed) DATAKON 2010 Proceedings. Mikulov CZ, Ostravská univerzita v Ostravě, S. 161–168, ISBN 978-80-7368-424-2
11. Sharp A, McDermott P (2009) Workflow modeling: tools for process improvement and application development, 2nd edn. Artech House, Inc, Boston, p 449. ISBN-13: 978-1596931923
12. Strommer M, Murzek M, Sommer M (2007) Applying model transformation by-example on business process modeling languages. http://publik.tuwien.ac.at/files/pubinf_4798.pdf. Accessed 15 Feb 2010
13. Van Der Aalst W, van Hee K (2004) Workflow management models, methods, and systems. The MIT Press, Cambridge, p 368. ISBN: 0-262-72046-9
14. Wegrzyn A, Karatkevich A, Bieganowski J (2004) Detection of deadlocks and traps in Petri Nets by means of Thelen's prime implicant method. Appl Math Comput Sci 14(1):113–121
15. Wohed P, Van Der Aalst WMP, Dumas M, ter Hofstede AHM, Russell N (2006) Pattern-based analysis of BPMN—an extensive evaluation of the control-flow, the data and the resource perspectives. http://www.workflowpatterns.com/documentation/documents/BPM-06-17.pdf. Accessed 15 Dec 2009
16. Workflow Control-Flow Patterns (2006) A revised view, BPM center report BPM-06-22. http://www.workflowpatterns.com/documentation/documents/BPM-06-22.pdf. Accessed 12 Dec 2009
17. Workflow Management Coalition (1999) Workflow management coalition terminology & glossary, United Kingdom. http://www.wfmc.org. Accessed 15 Jan 2010

Managing Step Changes in Information System Support: Lessons from an Industrial Study

Helena McCabe, David W. Bustard, Kevin Curran and Niall Byrne

Abstract In principle, information systems should be tuned to provide optimal support for the business processes of the organisation in which they are used. In particular, this means maintaining alignment between information systems and business processes as an organisation evolves. Where this strategy becomes difficult, however, is when an organisation reaches a point where a 'step change' in information system support is required. This paper examines how such step changes can arise and considers the factors involved in realigning the organisation when encountered. The discussion is illustrated through an industrial study in KTL, an Irish telecommunications service company. The analysis of the situation uses Soft Systems Methodology.

1 Introduction

For most organisations, progressive evolutionary growth is preferred to radical revolutionary change [1, 2]. The reasons are obvious: small changes can be accommodated without significant disruption and so carry less risk of a negative outcome. Unfortunately, however, not all change can be made comfortably in small steps [1, 3]. So, for example, although a restaurant can take on more kitchen and serving staff as its clientele increases, the choices are not so clear when its popularity means that it often has to turn customers away. Should it expand its premises (if practical), move to another site, or simply decide to be exclusive? Essentially, this may mean a *step change* for the restaurant—a change which has to be approached carefully to minimise the risk involved and ensure a successful outcome.

Such technological discontinuities are common to all evolving organisations and have been discussed in the literature over many years [4–6]. A particular concern of this paper is the point where expansion means that existing information systems

H. McCabe (✉)
KTL, Naas, Ireland
e-mail: Lena.McCabe@ktl.ie

D. W. Bustard
e-mail: dw.bustard@ulster.ac.uk

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_39,

need to change substantially. Broadly, the paper considers how best to recognise and manage step changes in organisations. The associated challenges and strategies are illustrated through reflection on the evolution of KTL, a telecommunications service company in Ireland. Section 2 of the paper gives a brief history of KTL leading up to its recognition of the need for substantial change. This is followed in Sect. 3 by details of an analysis of its circumstances and the requirement for change, identified through a Soft Systems Methodology analysis [7–11]. Section 4 then sets out and evaluates the change options from which a specific plan of action was developed. Section 5 concludes the paper with a consideration of the general lessons learned from the experience and an identification of future work.

2 Brief History of Company

KTL was formed in 1998, primarily offering construction services as a sub-contractor to the telecommunications and power industries. The company operated out of a private house, with just two permanent staff and others appointed on contract, as required. The only computing support needed was standard personal computers, with no requirement for a business website at that point.

Expansion soon followed; the first wave of growth was linked to a general industry trend towards client outsourcing, which KTL was well positioned to serve. By 2000, the company had moved to central Maynooth town. This was an easy decision as the business was prospering and an opportunity arose to acquire the premises concerned. Following the move, staff numbers increased to approximately 30, four of whom were in the office and the others located on site. In this period, most site documents associated with individual projects were handled in hard copy form and posted to the support team as necessary. In the office, however, there was a growing dependency on email, MS Word and MS Excel, creating a need for file management support.

With such a small office, outsourcing was the only realistic option and brought a number of practical advantages. In particular, outsourcing enabled KTL to draw on the expertise of the company involved and acquire the services it needed cost-effectively, with server monitoring being the main requirement initially. Managing such services directly would have required KTL to hold relatively expensive software licenses for applications that were only needed periodically. Drawbacks however, included the enforcement of formalised processes around informal day-to-day tasks; this required an ‘issue handler’ on KTL’s side, responsible for relaying issues back and forth to the IT Support Company. At times this has led to miscommunication and time wasting on both sides.

From 2000, KTL’s client base continued to expand, with blue-chip companies such as ESB and Vodafone bringing substantial work to the business. As a result, in 2002, KTL moved to modern offices on the Maynooth Business Campus. The decision to move was again straightforward as, in addition to gaining warehousing and storage facilities, there was fleet parking, which improved interaction among office, warehousing and fleet staff. The staff count was by then up to 50.

Expansion continued, putting progressively greater pressure on the information systems in use. A staff database was developed internally in 2005 (using MS Access) and became a comprehensive employee management system by 2006. In the same year, KTL launched its first website, and purchased its first enterprise-level software application. This was a Sage financial system used to track warehousing stock, purchasing information and other financial data. By this time, the company employed 130 staff directly, while also retaining an additional committed sub-contractor base of 50 people. KTL had also developed a comprehensive network of six regional offices and warehouse facilities, investing €5 m in plant, equipment, and a large fleet of vehicles.

From its establishment in 1998, KTL had made several step changes in its use of premises but had maintained an evolutionary approach to information system support. As a result, by 2005–2006, the MS Office base that was adequate for the growing start-up was no longer suitable for the medium sized enterprise that KTL had become. Ideally, this would have been the time to consider a step change in technology support, reviewing needs across the whole organisation but the company instead continued to address individual issues as they arose.

The next pressure point emerged when the office staff felt they could no longer support the site crews through manual processes. Various inefficiencies were identified. For example, up to then, site crews were required to take photographs on completion of a job and post camera film to the office for subsequent delivery to a print shop. This spurred a major upgrade of the IT function in 2007, with all project managers and crew leaders supplied with laptops, digital cameras, mobile internet connections and mobile phones. The initiative also included setting up secure remote access, full disaster recovery and robust back-up capabilities. This development enabled site crews to deal with all documentation and supporting photographs digitally, which greatly improved business performance.

By 2009, following yet more expansion, the difficulty of managing a large number of individual complex jobs became problematic. Again a specific IT solution was sought and KTL used the government-backed InterTradeIreland FUSION Scheme in Ireland[1] as a way of facilitating the necessary change. In practice, however, the project brought the wider benefit of reviewing the overall information system needs of KTL to help facilitate what turned out to be a much-needed step change in technological support.

3 SSM Analysis of KTL

InterTradeIreland FUSION is an all-Ireland cross-border programme that enables knowledge and technology transfer between business and academia to support innovation and increase capability. The project with KTL had the University of Ulster as an academic partner and provided funding for the appointment of an Information

[1] http://www.intertradeireland.com/index.cfm/area/information/page/FUSION.

Systems Analyst for 18 months. The overall goal of the project was to design and implement information system support for project management and business processes within KTL.

The project began formally in July 2009, following the appointment of the Information Systems Analyst. An initial meeting was arranged to review the detailed objectives and work plan. This took the form of a workshop, which followed an informal participative approach to building a shared understanding of the 'problem situation' among those present. Soft Systems Methodology (SSM) was used as the method of analysis to help build an appreciation of the wider factors that might have an influence on the progress of the project.

The first stage of analysis involved the creation of a Rich Picture diagram [12] as shown in Fig. 1. This was built up step-by-step, identifying individual business functions and their inter-dependence, and documenting any concerns expressed. The rich picture diagram brought out issues directly relevant to the project, including:

- Capacity for email and data storage in the company had become critical; the server appeared congested with large amounts of project related documents.
- Staff felt burdened by the amount of paper work and administration required to fulfil their role.
- Multiple versions of documents were circulating simultaneously.

These findings were reinforced, later in the analysis phase, through conducting a company-wide staff survey. This also gave staff an opportunity to identify additional problem areas in their individual work.

As each area of the Rich Picture diagram was discussed, it became clear that there were significant opportunities for improvement in most aspects of information systems provision. In effect, the company had been delaying decisions on the introduction or enhancement of information system support for several years to a point where a step change was now needed.

In KTL, the need for a project management information system had become critical so there was pressure to move that aspect of change along as quickly as possible. To help cover the wider analysis to some extent, however, a high level business activity model was created for the company, documented as an SSM conceptual model, as shown in Fig. 2. This identifies the main activities in KTL's service business, which is essentially based on securing individual projects through competitive tendering. The focus of the FUSION project was on project management, represented by activity *8. Complete Awarded Projects* in the figure.

From an information systems perspective, the goal was to help maximise efficiency when completing awarded projects. The associated changes included:

- Streamlining business processes through workflow technology, thus consolidating similar processes, partially systemising manual processes and improving process continuity and quality assurance
- Reducing administrative workloads through automating manual processes within the company, including the normalisation of data input, auto-populating data input and auto-generating document templates, while automating document content

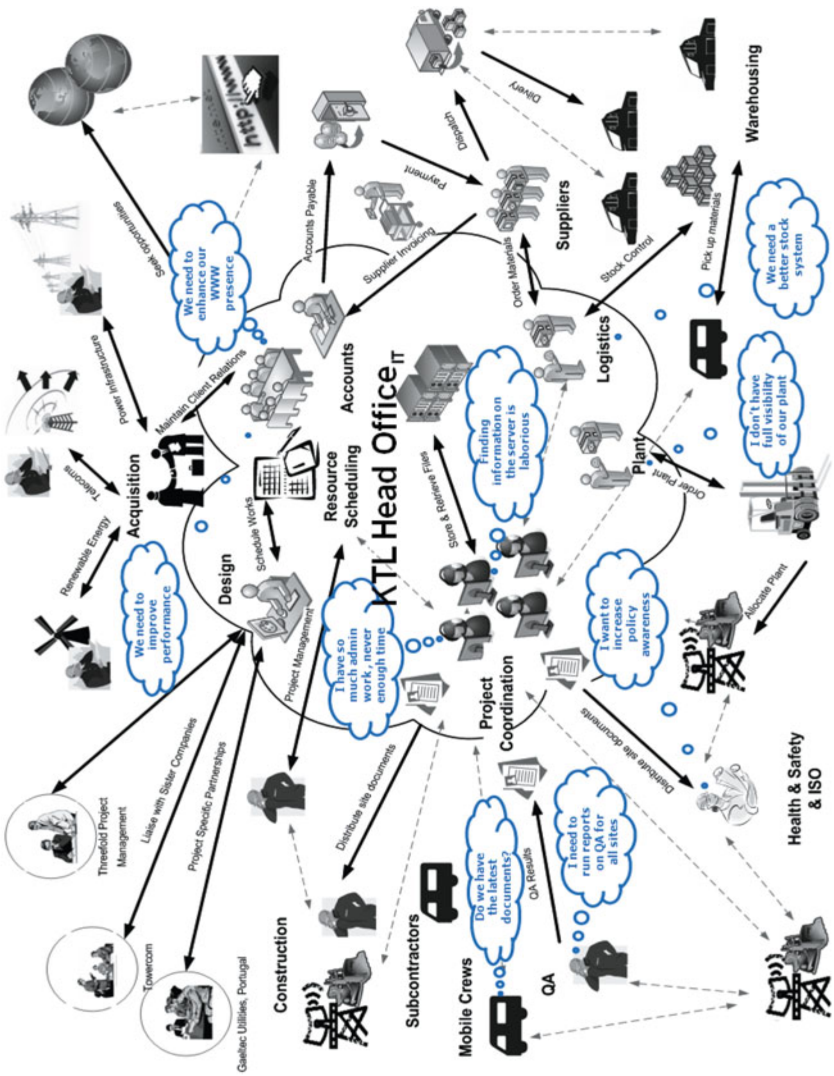


Fig. 1 KTL rich picture diagram

- Improving access to information through the provision of database-driven technology, supporting advanced filtering and data searching
- Increasing company-wide collaboration by offering remote access to enable dispersed projects teams, located on site and in the office, to have the same central access to share documents, tasks, and calendars
- Facilitating document management via a centralised repository

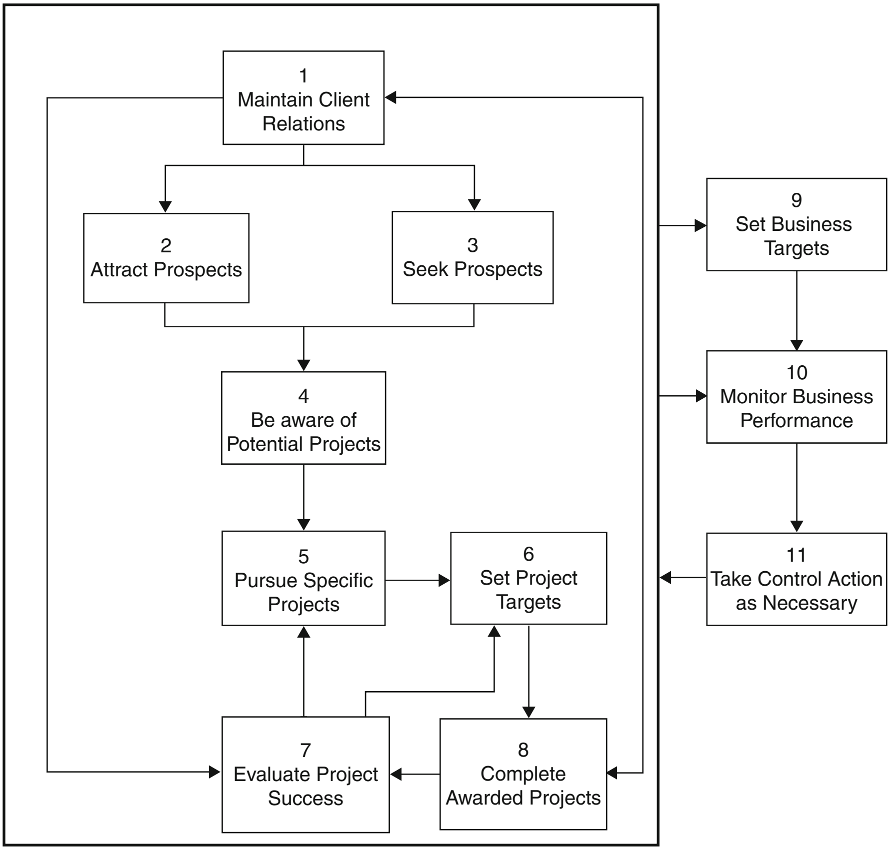


Fig. 2 Top level business activity model for KTL

The analysis phase appeared to isolate the project management system (PMI) to a reasonable extent but further problems were encountered at the next level of investigation: the technology evaluation phase, as discussed in the next section.

4 Change Options and Action Plan

Having broadly identified the project management information needs of KTL it was then necessary to consider the technology base on which it would be developed. Five options were identified, as summarised in Table 1.

In essence the choices ranged from commissioning the development of an enterprise-level system, through tailoring an existing framework that provided most of the necessary functionality, down to implementing a system from basic technology components. There was also an opportunity to buy a system that a similar

Table 1 KTL project management information system options

Type	Description	Examples
1 High-end business process management solutions	Enterprise-level, highly customisable systems that support all business functions (Finance, ERP, HR etc)	SAP, Microsoft Dynamics
2 Mid-range boxed products offering project management solutions	Usually purchased as software as a service, offering out of the box functionality	Maconomy Project Management, Basecamp, Hyperoffice
3 Mid-range semi-boxed products	Products that offer base functionality but also enable customisation to meet business needs	SharePoint, Interneer Business Process Management
4 Build code from scratch	Coding a new application. The benefits include omitting the need for vendor involvement and gaining ownership of IP	ASP.Net application, C# code, SQL server database
5 Purchase code from software development company	An opportunity arose to purchase code from a software company that developed a similar system	Existing bespoke system

company had developed in-house. Unfortunately, all of these options had significant drawbacks so there was no obvious best choice. Specifically:

1. For an enterprise system the costs and complexity of change were significant.
2. The mid-range boxed products examined didn't allow enough customisation.
3. In the mid-range semi-boxed products area, SharePoint seemed attractive initially but customisation effort and associated costs were a problem.
4. Developing a new bespoke system from scratch was unrealistic with the limited staff time available. Moreover, it would be very difficult to estimate the effort and costs associated with this approach, and it was considered inappropriate to develop new software in an area where there were many products on the market.
5. Buying an in-house system from a similar company also seemed attractive initially but the costs were high and some customisation was still required, implying a need to develop a detailed knowledge of the implementation.

Despite the pressure to make a decision quickly, further analysis was performed on the 'least-worst' option of building on a semi-boxed product that could be customised. At the same time, lower level details of project management processes were mapped out. This work identified additional difficulty in that there were stages in the end-to-end project lifecycle that were tightly coupled with financial processes, making it necessary to integrate with the financial management system. To determine the best way to link with the new financial system, being provided by Sage, the vendor was given a copy of the project management requirements and invited to comment. Surprisingly, the Sage web-based Customer-Relationship Management (CRM) product was recommended. Certainly this had the necessary interoperability with the Sage finance system but seemed inappropriate, as its role was apparently quite different. However, despite how it was marketed, on further investigation it did indeed meet all requirements. Sage CRM includes comprehensive business process management capabilities and also supports substantial customisation. As well as being flexible, it is also a mature product from a mature organisation, and purchasing both products from the same vendor would reduce the cost of initial provision and subsequent support. Thus, the Sage CRM platform was proposed to senior KTL management as the most viable solution for the PMI project, and procured in January 2010.

Further analysis noted a link with the HR system and a need for it to be made an enterprise-wide application. The resulting architecture is shown in Fig. 3. The delivery of these three major mid-enterprise level applications (PMI, FMS and HRMS) is now integrated within what has become known as *Project Fusion*. A phased roll-out plan covering all three applications has been developed, as implied in Fig. 3. The main objective of integration is to avoid duplication of data, meaning that shared data need only be entered once, through one application, with changes reflected automatically in the others. Such integration is needed when systems share concepts or entities. Shared entities between the PMI and FMS system include: projects, jobs, clients, suppliers and project-related financial information. Shared entities between the PMI and HRMS system include: employees, resource teams, leave planning and training certification.

At the point of writing (June 2010), Project Fusion had the following status:

- Phase 1 of the FMS was successfully in place, supporting all essential financial activities
- The HRMS was in the product evaluation phase
- The PMI was under development and due for launch in September 2010
- Phase 2 of the FMS project, which included the implementation of a new stock management system was due by the third quarter of 2010

While the project has been relatively successful so far, problems remain. For example:

- Deploying a new FMS resulted in changes to current business processes. These changes were not fully appreciated by business users who then requested incompatible design requirements for the PMI system or failed to appreciate some of the implications involved.
- Ideally, each new system would have been introduced and tested individually. In practice, the pressure for change meant that several inter-connected systems were changed simultaneously, which increased the number of errors involved and made such errors more difficult to locate.
- Having a systematic, well-designed, well-documented approach to system change raised management expectations to too great an extent. There still remain significant difficulties in gaining consensus and commitment for the enormous effort that is required. Issues include ensuring adequate resource allocation, obtaining

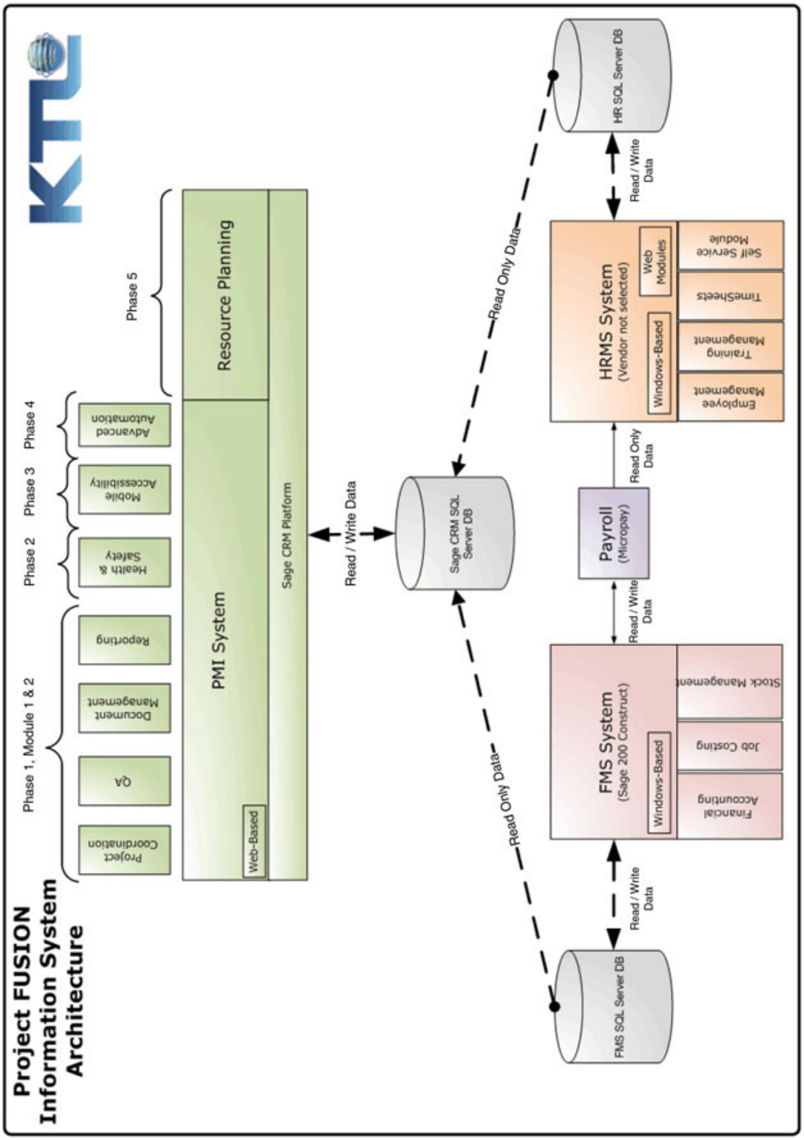


Fig. 3 KTL information systems architecture

timely budgetary approval and achieving ongoing commitment to business change at all levels of the organisation.

Project Fusion now has the full backing of senior management, as recognised by the introduction of a weekly review meeting attended by the KTL Board of Directors. The company is also investigating the potential employment of technical staff, in addition to the FUSION Information Systems Analyst, implying the establishment of an internal IT Department to complement the existing outsourced IT support.

5 Conclusion

This paper has highlighted two general issues in organisations that require further research: the effective handling of step change in information systems and the effective management of information systems to minimise the need for such change. These research problems have emerged from a specific industrial project but the situation investigated seems common to all expanding companies. Indeed the step change problems seem to be a significant risk for any organisation in trying to keep pace with frequent and substantial changes to available technology.

The project has helped clarify the symptoms that indicate the need for a step change. In particular, these include:

- Significant problems appearing in a number of information systems simultaneously
- The best course of action being unclear, with all apparent options have significant drawbacks, such as high cost or significant disruption to existing work
- Staff working longer to overcome deficiencies
- Staff becoming so pressured by day-to-day activities that they have little time to contribute to the definition and implementation of desirable change
- A tendency for management to delay change, waiting to react to difficulties, rather than taking a proactive approach to improvement
- Some deficiencies becoming critical, prompting urgent action to implement change
- Little time available to identify and analyse change options when action is needed, with resulting changes having a greater risk of failure

The project was fortunate in taking an SSM approach to the analysis of the initial situation which focused on the narrower goal of developing a project management information system (PMI). SSM brought out the wider issues involved, leading to the conclusion that an holistic approach to change was needed. In effect, this is simply a confirmation of the importance of taking a broad systems approach to change, in general, and acknowledging the value of SSM in supporting such an approach. What remains, however, is the need to raise the profile of this general issue to help organisations recognise the risks involved in step change, adopt practices that manage such risks and give guidance on how best to approach such change when necessary.

For example, the possibility of a step change being needed increases whenever there is any delay in change. This can occur, for instance, because the options available are unattractive, the cost of change is significant or the work involved seems disruptive to normal business.

Further research is needed but the main lessons learned so far are:

- Organisations can benefit significantly from developing a good understanding of the relationship between their business processes and the information systems that support them. This seems true of organisations of any size. The implication here is that they maintain an enterprise model and use it in identifying and facilitating change. Again, SSM seems valuable in supporting such analysis but whatever technique is used it must be well understood across the organisation and be relatively straightforward to apply.
- Organisations require processes that monitor the gap between ideal information system support and what they currently have in place. As the gap increases so also does the risk of step change being required. Delay may be desirable, however. This may occur, for example, when a situation is volatile or when revenue generating activities are given precedence over maintenance and strategic planning. Indeed, it can also be argued that there is some advantage in imposing delay as it helps ensure the validity of the assumptions underpinning any planned change. Delay may also be beneficial if costs are expected to fall. The important point here, however, is that delay should be *deliberate*, with the implications of handling a step change fully understood.
- When a step change is needed there is a greater than ever need to analyse the situation carefully to determine the best course of action. This can be difficult as there is likely to be significant pressure at that point to take a decision quickly. This suggests a need for more scenario planning and greater risk management in general.
- Step change is inevitably disruptive so the staff affected should be given due consideration in recognising the pressures building up to the change and the commitment needed to implement the change smoothly.

Much has been learned from the study at KTL and as the project continues, the intention is to help the company develop a way of working that makes the handling of step change routine. This is particularly important as KTL has a strategy of adapting quickly to changing market conditions, reflected in its recent expansion into the sustainable energy sector. Indeed KTL is currently going through a fourth wave of growth involving another move to larger premises. This has put even more pressure on ensuring that the plans for information system change are robust and aligned with the organisational change. The work is being pursued in a way that enables the results to be documented and shared.

Acknowledgements This work was completed through an IntertradeIreland FUSION project between KTL Ltd and the University of Ulster.

References

1. Mayle D (ed) (2006) Managing innovation and change, 3rd edn. Sage (with Open University), London
2. De Bono E (1971) Lateral thinking for management. Penguin Books, London
3. Hammer M, Champy J (1993) Re-engineering the corporation. Nicholson Brearley Publishing, London
4. Foster RN (1985) Timing technological transitions. Technology in Society, Special Issue: Technology in the modern corporation: a strategic perspective 7(2–3):127–141
5. Tushman ML, Anderson P (1986) Technological discontinuities and organizational environments. Admin Sci Quarter 31(3):439-465
6. Phaal R, Farrukh CJP, Probert DR (2004) Technology roadmapping—a planning framework for evolution and revolution. Technol Forecast Social Change 71:5–26
7. Checkland P, Poulter J (2006) Learning for action: a short definitive account of Soft Systems Methodology, and its use by practitioners, teachers and students. Wiley, New York
8. Checkland P (1999) Systems thinking, systems practice. Wiley, New York
9. Checkland P, Scholes J (1999) Soft systems methodology in action. Wiley, New York
10. Wilson B (2001) Soft systems methodology: conceptual model building and its contribution. Wiley, New York
11. Wilson B (1990) Systems: concepts, methodologies and applications. Wiley, New York
12. Lewis PJ (1992) Rich picture building in the soft systems methodology. Eur J Inf Syst 1(5):351–360

Using Agile Practices to Build Trust in an Agile Team: A Case Study

Orla McHugh, Kieran Conboy and Michael Lang

Abstract Trust is an important aspect of any software development team, but particularly with self-managing teams as team members are very dependent on one another. Agile teams are considered to be self-managing and they employ many different agile practices to function as an agile team. While there have been many studies of trust in software development teams few have examined trust in an agile context with even less focus on how specific agile practices may contribute to trust. The purpose of this study is to examine how three agile practices—the daily stand-up, iteration planning and iteration retrospective—may support and facilitate trust in an agile team. An exploratory case study of one agile team was conducted. The findings indicate that while factors such as environmental conditions and personal characteristics of team members must be considered, agile practices can also contribute to building trust among team members. They may also highlight the existence of a lack of trust.

1 Introduction

Agile software development (ASD) refers to a group of agile methodologies that focus on developing software in short time periods (iterations). They allow requirements to evolve and change during iterations, encourage close collaboration between agile teams and users, and have teams that are self-organising and cross-functional (Agile Alliance 2001). ASD has evolved since the mid-1990s and there are now many different agile methodologies in existence such as eXtreme Programming (XP), Scrum, Dynamic Systems Development Method (DSDM) and Feature Driven Development (FDD). These methodologies are often called "lightweight" methodologies as they differ in their approach to the traditional predictable "plan-driven" method of developing software, which requires software teams to follow many processes, and to produce lots of documentation (Boehm 2002).

O. McHugh (✉)
Business Information Systems, J. E. Cairnes School of Business and Economics,
National University of Ireland, Galway, Ireland
e-mail: orla.mchugh@nuigalway.ie

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_40,

The Agile Manifesto places great emphasis on the agile team and the role of the individuals within the team. Teams should be self-organising and self-managing, contain motivated individuals, be provided with the environment and support they need, and be trusted to get the job done (Agile Alliance 2001). With ASD the team is provided with substantially more control than it would have had when using a plan-driven approach to software development. This is a dramatic change for the project manager, who has traditionally been the primary controller (Nerur et al. 2005). Project managers now need to place great trust in their team members to make the right decisions and complete their tasks in a timely manner. One way of ensuring that this may occur is through the various agile practices that are used by the agile team.

1.1 Research Objective and Motivation

Research in the area of ASD has grown in recent years due to the increase in the number of software project teams that use an agile methodology (Abrahamsson et al. 2009; Conboy 2009; McEvoy and Butler 2009). Each agile methodology details various practices that distinguish it from other agile methodologies, but they each follow the same underlying agile principles (Agile Alliance 2001) where a practice can be described as a "common way of acting", which is accepted by a group of individuals as the "correct way to do things" (Hansson et al. 2006). Agile teams can choose to adopt the agile practices that suit their environment or that work well for them, bearing in mind that these practices may span several agile methodologies (Elssamadisy 2007; Hansson et al. 2006). These agile practices may be technical (e.g. test driven development, continuous integration), relate to planning (e.g. iteration planning, daily stand-up), or could relate to the agile environment (e.g. co-located team, self-organising team).

The objective of this study is to explore how trust amongst agile team members can develop and be nurtured as a result of using agile practices, which in turn may develop positive team outcomes such as fostering better relationships/cohesiveness amongst team members or improved team performance. Previous studies highlighted the importance of trust in agile teams (Das and Teng 2001; Mayer et al. 1995; Nerur et al. 2005), but little has been said about how the use of agile practices can increase or decrease trust among team members, which is a motivation for this research. There have also been recent calls for further research that is more practice-focused (Dybå and Dingsøyr 2008) and to investigate how each distinct agile practice can help to optimise the performance of an ASD team (Maruping et al. 2009). Consequently, three practices were selected for the purposes of this study (see Table 1), on the basis that they are amongst the more commonly used agile practices by practitioners (Version One 2009). Each of these practices is related to the management and control of an agile project and requires the collective participation of all team members with a focus on people, communication, interaction and teamwork.

Table 1 Agile practices studied (Beck and Andres 2005; Elssamadisy 2008, Schwaber and Beedle 2002)

Agile practice	Description
Daily stand-up	The daily stand-up is a short daily status team meeting lasting a maximum of 10–15 min typically conducted at the same time each day. The meeting is conducted with team members standing up. During the meeting team members explain briefly what they accomplished since the previous meeting, what will be completed by the next meeting and indicate any impediments that may prevent them from completing these tasks
Iteration planning	The iteration planning session is a meeting that takes place at the start of each iteration where the team collectively define and plan tasks that must be completed during the next iteration
Iteration retrospective	An iteration retrospective is a meeting that is held at the end of each iteration where the project team reflects on what went well in the iteration, what did not, and what could be improved for future iterations

The remainder of this paper is structured as follows. Section 2 provides an overview of the literature on teams, agile teams and trust and then introduces the research question. Section 3 provides details on the case organisation. Section 4 details the methodological approach for this study. Section 5 presents the findings from the case study and the final section discusses the findings, details the limitations of the research and outlines recommendations for further research.

2 Background

2.1 Teams

Teams are groups of individuals that work together, are dependent upon one another and have one or more tasks to perform in order to accomplish various goals (Hackman 1990; Mayer et al. 1995). Teams should comprise of individuals who are technically competent, are productive, committed to the team, and have good problem solving and interpersonal skills (Jurison 1999). There is also value in ensuring that a team has a mix of personality types, both introvert and extrovert, which can lead to a more successful team (Jurison 1999). There are many conditions that must be met in order for teams to be effective such as: creating a team that can work well together; ensuring the team are committed to the organisation; providing the team with autonomy to make decisions; and creating a supportive environment that provides the team with all the necessary resources and skills in order for them to conduct their work (Wageman 1997; Wageman et al. 2009).

Teams can be manager-led or they can be self-governing and self-managing (Hackman 1990). Self-governing teams set their own goals, select new members, and manage and execute work of their own design (Hackman 1990).Self-managing teams are teams that have responsibility for managing their own work and behaviours but, others usually make decisions about goals, team structure, and organisational supports (Barker 1993; Cohen et al. 1997; Manz and Sims 1987). Both types of teams are empowered and have autonomy to make decisions about their tasks and the processes that they use, which are traditionally the responsibility of supervisors and managers (Alper et al. 1998; Cummings 1978). To perform well as a team all members must be committed to the team and must feel that they have the support of other members (Bishop et al. 2000) as the relationship between individuals within teams can impact on the dynamics of the team (Gruenfeld et al. 1996).For example, teams of individuals that are more familiar with each other may be more effective at sharing information and views than those who are not (Gruenfeld et al. 1996).

2.2 Agile Teams

Agile teams are considered self-managing and self-governing (Cockburn and Highsmith 2001). Yet, it cannot be assumed that by putting a group of individuals together in a team and calling them "self-managing" means they are automatically agile (Moe et al. 2010). While the optimal size of an agile team has been debated, ASD teams are typically small with no more than ten team members (Schwaber and Beedle 2002). Team members should have a range of skills, be cross-functional and have the ability to complete the required tasks (Elssamadisy 2008). A team must be empowered to make decisions and is responsible for meeting the goals of each iteration in whatever way it deems appropriate (Schwaber and Beedle 2002). However, the team must conform to any existing standards within the organisation such as coding standards, hardware/software platforms etc. (Schwaber and Beedle 2002).

To ensure a team produces quality work an appropriate and supportive environment must be available to team members, for example, ensuring availability of required tools, and open-office space to facilitate open communication. There is also a necessity for team members to be cooperative, collaborative, trusting, have good relationships with each other, and be able to make decisions quickly (Cockburn and Highsmith 2001). It has been questioned whether agile methodologies are suitable for a distributed team for many reasons including the possibility that distributed team members are less likely to feel part of the same team as co-located team members (Ramesh et al. 2006). These can be alleviated somewhat by site visits, the facilitation of collaboration and knowledge sharing and supplementing informal communication with documentation (Ramesh et al. 2006).

2.3 Trust

Trust has been studied in many different contexts, yet there is little agreement on a single definition with the term used in many different ways (Blomqvist 1997; Kramer 1999; Lewicki et al. 1998; McKnight et al. 1998; Rousseau et al. 1998). One such example is that presented by Lewicki et al. (1998) who states that "trust is the confident positive expectations regarding another's conduct (words, actions and decisions)". A second definition by McKnight et al. (1998) define trust to mean that "one party believes in, and is willing to depend on, another party", which is of great importance in an environment where individuals must interact and work together to fulfil a common goal. Trust, or a lack of trust, can exist such between individuals, groups and organisations (Das and Teng 2001) with trust fostering cooperation amongst parties (Rousseau et al. 1998) and a lack of trust causing suspicion or a lack of confidence in other parties (Kramer 1999).

As organisational teams become more diverse with team members from a variety of backgrounds and culture, the development of trust between all members is extremely important for them to work together effectively (Mayer et al. 1995). Individuals with different personality types, experiences and cultural backgrounds vary in propensity in how likely they are to trust others (Hofstede 1980) with levels of trust evolving or diminishing over time as they interact with each other and observe each other (Das and Teng 2001; Mayer et al. 1995). Distributed teams face other challenges such as lack of control, lack of cultural understanding, miscommunication, limited opportunity to communicate orally due to time differences and lack of team morale and trust between team members (Ramesh et al. 2006).

The emergence of self-managing agile teams increases the importance of trust among team members as members are relatively free to develop the processes they prefer and to set targets they consider appropriate (Das and Teng 2001; Mayer et al. 1995). Team members that collaborate and trust each other are imperative for the success of an agile project, which may be difficult for developers who are used to working predominantly on their own (Nerur et al. 2005). Individuals or teams must believe that each individual within the team has the ability, knowledge, and competence to complete the tasks required and they must also have high personal and moral integrity (Mayer et al. 1995). Therefore, it is important to maintain and strengthen trust between team members.It may take some time and effort for an organisation to build a culture of trust amongst team members (Nerur et al. 2005), but it is possible that this may be facilitated and supported by the use of agile practices. Prior research in this important area is limited, which leads us to the following research question:

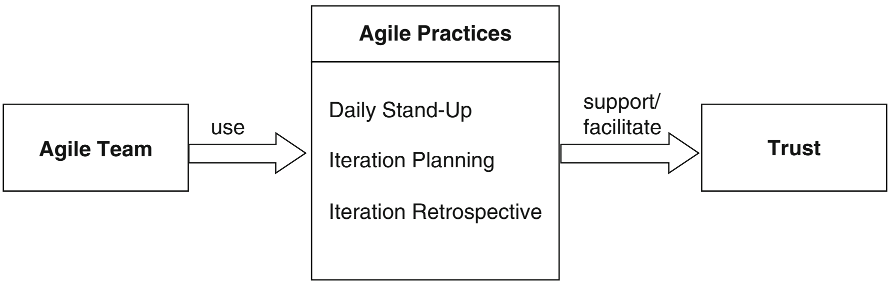


Fig. 1 Research model

How do agile practices contribute to building trust in an agile software development team? To answer this research question (see research model in Fig. 1) a case study approach is used where a single agile team is studied. This research is part of ongoing research for a Ph.D. and to date data has been collected from one team. At this point this research is exploratory and data will be collected from several other teams in the near future.

3 The Case Organisation Studied

The organisation selected for this case study is a large multinational financial services organisation with offices located worldwide. A decision was made by the Head Office in the United States to introduce a customised version of an agile methodology across the organisation. The Research & Development (R&D) division in Ireland was the first within the organisation to pilot the use of an agile methodology for developing software. The project studied is a long-term project that was in existence for 2 years at the time of the study and it is envisaged to continue for at least another year. The project involves the development of a set of back-end Web services that are used by various front-end applications for developing financial analysis documents. The end users are financial analysts across several business units, although the direct customers are the IT groups in six different business units who develop the front-end applications. This results in a number of different customers, all whom have competing needs.

The team composition has changed intermittently since its inception. When data collection commenced the team was composed of ten individuals distributed between the United States, Ireland and India. The development team was primarily based in Ireland with the Quality Assurance (QA) function based in India and a database specialist and customers based in the United States. The customer was not directly involved in the project on a day-to-day basis with the analyst acting as the proxy customer. Shortly after data collection commenced, the two QA team members based in India, who were both testers, departed from the team. Recruitment of two replacement team members is currently taking place, but these will be now based in Ireland. All but one of the team members interviewed is very experienced, with most of their experience obtained in a non-agile environment. A profile of the team at the start of data collection is detailed in Table 2.

Three week iterations are used by this agile project team. The iteration planning meeting and iteration retrospective meetings are generally combined into one meeting (approximately 1 h in duration) with distributed team members available on a conference call. The first 15 min of this meeting consists of the iteration retrospective. The project manager asks team members in turn to briefly comment on their work in the previous iteration, to indicate what went well and what can be improved for future iterations. The remainder of the meeting focuses on iteration planning where tasks (user stories) for the next iteration are agreed upon, and estimates are reviewed. Team members are aware of the user stories that are assigned to them prior

Table 2 Team profile

Role	Software development experience	Experience in organisation	Experience in the agile team	Location
Project Manager [P1]	13 years	8 years	3 years	Ireland
Analyst [A1]	15 years	2.5 years	2 years	Ireland
Database Specialist [DB1]	15 years	5 years	11 months	USA
Developer [D1]	10 years	3 years	3 years	Ireland
Developer [D2]	10 years	3 years	6 months	Ireland
Developer [D3]	12.5 years	2.5 years	1.5 years	Ireland
Developer [D4]	2.5 years	2.5 years	2.5 years	Ireland
Developer [D5]	10.5 years	4.5 years	1 year	Ireland
Tester [T1]	Departed team	Departed team	Departed team	India
Tester [T2]	Departed team	Departed team	Departed team	India

to the meeting and each team member will have prepared a time estimate for each task.

Daily stand-ups are held 4 days a week as on the 5th day a project team meeting is held instead of the daily stand-up meeting. Daily stand-up meetings generally last 10–15 min and take place in an office as this facilitates a conference call with distributed team members. As with the iteration retrospective, the project manager, or senior developer, if the project manager is not present, directs the meeting. All team members in turn briefly comment on what has been completed since the previous daily stand-up, what they are currently working on and whether there are any "blockages" inhibiting the completion of a task. In the event that further discussion is required in relation to a task(s) this will typically take place amongst affected team members following the completion of the meeting.

4 Research Design

Case studies are particularly appropriate for exploratory research which is at an early stage of maturity (Benbasat et al. 1987). Access to the team in question was readily available, and it was felt that the opportunity should be utilized to conduct a qualitative study and gain an understanding of the agile practices in more detail. An interview guide was developed from the literature on teams, agile methodologies, and trust. It predominantly contained open-ended questions as this provided the researchers with the opportunity to ask additional questions (Cooper and Schindler 2001). Sample interview questions are available in the Appendix.

Data collection took place over a 4-month period. Each interview followed a similar structure. Details on the project and on the number and type of agile practices utilized by the team were gathered from the first interviewee only. It was felt that it was only necessary to ask these questions once and it would minimize the amount of time required to interview the remaining participants. All interviewees provided details on their background, and level of experience, and this was then followed with a number of questions under different headings that were asked to each interviewee.

Semi-structured interviews were used to capture the responses of each of the eight participants. This gave the interview a structure, but also allowed thoughts and personal experiences to emerge from participants. Seven of the eight interviews conducted were face-to-face interviews. The remaining interview was conducted using a conference call as this individual was based in the United States. The two individuals in India were part of the team when a number of interviews were conducted, so references are made to these in the findings even though these team members were not interviewed. All interviews were conducted at the offices of the organisation. The interviews lasted approximately 1 h each. All interviews were recorded and transcribed.

The transcriptions were reviewed, coded and categorised based on the interview guide. The categories were then sub-divided into further categories in order to identify patterns and themes and to validate the data from different individuals (Miles and Huberman 1999). The findings were analysed and validated by cross-checking the findings with each of the other participants.

5 Findings

The findings are now presented detailing how the agile practices have impacted positively and negatively on trust in an agile team.

5.1 Team Composition and Trust Amongst Team Members

The team studied is predominantly a well-established, experienced, self-organising team with team members appearing to have a good work ethic and track record of delivering on what had been promised. Team members are very collegiate and supportive of each other and appear to work together as a unit with many items discussed collectively. The team *look out for each other and make sure that when something goes wrong… it is very much a team effort to fix it [D5].*

A lot of trust exists between team members. All team members believe that their colleagues are competent and can complete the tasks allocated to them. Individuals work on designated tasks becoming experts in a particular area. As a result, team members trust and accept that their colleagues are honest when determining estimates and can be believed when they say that a task is complete.

The project manager does not micro-manage and trusts team members to *work a good solid day [D1].* He also trusts the team to accurately define estimates for tasks as *they [developers] will have more context than me [P1]* and to then deliver on those estimates. It is rare for other team members or the project manager to question a time estimate: *planning estimates aren't really collective decision they're just presented* *by developers as their individual times and agreed [D1].* This was corroborated by the project manager who *doesn't tend to take a lot of decisions [P1].* Instead, decisions are made by the team.

5.2 Building Trust with Agile Practices

All three agile practices are an important component of building trust in an agile team. In particular, the stand-up is a daily touch-point for all team members, which requires team members (co-located and distributed) to meet and communicate with each other on a daily basis and *keeps the lines of communication open [D4].* Speaking to each other on such a regular basis improves communication, helps individuals to better understand each other, become familiar with their personalities and traits and be more comfortable in their interactions with each other leading to increased levels of trust. At the start of each meeting there is usually some banter between the different team locations, which *probably helps you to develop a relationship there that you are not really aware of [D3].* It also *sets a good tone [D4]* for the meeting. This interaction helps to build a rapport and trust between team members over time. The practices *help with understanding people... as the more that is spoken frequently ...a bond kind of builds up [D5].* They have also resulted in a *good culture of just picking up the phone [D4]* to ask another team member a question or speaking to another team member informally outside of meetings. Team members do not feel that they need to wait for the daily stand-up to take place in order to discuss a problem and trust that their colleagues will provide them with the required support.

All three practices provide an open forum for knowledge sharing, transparency and feedback where *the information sharing is important. It's good to know you can be frank, throw your ideas out there [A1].* These practices help to build trust amongst team members because *they are having it [meetings] on a more regular basis [P1]* and *people get more comfortable [speaking] over time [D5].* Individuals are encouraged to voice their opinions in all three meetings without fear of repercussions and no-one has ever been *reproached for expressing an opinion [D4].* If a task takes longer than estimated the individual is not reprimanded, nor looked on negatively, but instead is asked *did we do the estimate wrong, or have you got something that is blocking you? [A1]* and is helped to complete the task. The meetings provide team members with an *opportunity to question and once you get a valid answer back...,well then that does help [with trust] [A1].* If the environment was not as supportive there may be a tendency for individuals to *become more conservative when [they] plan [D4]* so as to avoid negative repercussions, which could *be very detrimental [D4]* to the project.

In particular, the agile practices have helped to alleviate the possibility of distrust that can be experienced with distributed team members from different cultures. As the team is distributed across three different cultures it can be difficult for team members to build good relationships with other distributed team members, especially when *you haven't met face-to-face [D4].* Participation by the QA (Indian) team in the daily stand-up and the planning/retrospective meetings helps to build trust with them as the QA team have a tendency to refrain from being too vocal and are *fairly timid kind of guys, they don't really say much other than "his is what I did" and "this is what I'm going doing today" [D2].* This may be *a cultural thing or may be the lack of experience in the team [D2].* This is of particular importance to the project manager who has had some trust concerns with the distributed team, but the *stand-up is a great way to keep on top of it [progress] [P1].* The team *know very quickly [P1]* of any actual or potential delays, which can be addressed immediately. A lot of conversation takes place*that wouldn't happen if these practices weren't being used [P1].*

The project manager believes that these practices have helped with trust as *anything that encourages conversation between people is going to build up a level of trust between developers [P1].* The daily stand-up in particular has made it easier for distributed team members to feel part of the team *because of the continuous communication between the team, it helped me feel part of the team [DB1].* This is particularly important with distributed team members as difficulties were experienced on several occasions with the QA team in India where *they chop and change them [team members] regularly [D2]* resulting in the regular re-building of trust and relationships. New team members also integrated faster into the team and quickly built trust levels with other team members because they were required to participate in the daily stand-ups, iteration planning and retrospectives from the outset.

5.3 Impediments to Building Trust

Team members do not consider the customer part of the team and there is a lack of trust between the customer and the team. There are a number of reasons for this. The customer does not partake to any great extent in any of the three agile practices, even though they are invited to participate. Therefore, regular communication between the team and the customer is limited. *Response times from the customer are very slow... it can be hard to get their time... there can be misunderstandings... they have their own agenda [D2]* and it can *get to the stage where you [the team] may have a chat with the customer and they would ask you to do X, Y and Z... and we get them to send an email so we have it in writing [D3].* The team do not trust that the customer will review releases of the software in a timely manner and *it's frustrating to go back there [to the customer] if there is nothing happening [A1].* Regular participation and interaction by the customer could benefit the team and increase trust between the parties. This was demonstrated to some extent when a site visit took place by a team member to the United States *as it encouraged a lot more [face-to-face] conversation [with the customer] [D1]* and was of great benefit to the team.

While the agile practices have had a positive impact on building trust and relations with distributed team members they could be improved further. Lack of face-to-face communication and regular site visits have been shown to increase levels of trust amongst distribute team members (Ramesh et al. 2006). In this organisation these could be alleviated to some extent through the use of technology such as video-conferencing, which is available for use within the company.

6 Discussion and Conclusion

These findings contribute to the literature on trust and agile teams by attempting to provide some insight into how daily stand-up meetings, iteration planning and iteration retrospectives contribute to building trust in an agile team. Overall, the team studied have reported a predominantly positive view of these three agile practices. Regular usage of the agile practices has resulted in the development and fostering of relationships and increased levels of trust between all team members that may not have existed otherwise. While the agile practices are not the only factors that help to increase trust in the team, they are acknowledged to be a contributing factor. They can also highlight where there is a lack of trust between individuals due to lack of participation in the agile practices.

It is important for an organisation to build a culture of trust among team members (Nerur et al. 2005). The co-located team in particular trust each other very much, which may be due to a number of different factors. The team studied was very experienced and cohesive. They are very committed to the team and are supportive of each other, all of which helps to build trust (Bishop et al. 2000). The majority of the team were co-located sitting in very close proximity of each other, which facilitates face-to-face communication and allowed team members to participate in or contribute to informal conversations, if necessary. However, the level of trust between the team and the customer, who is typically part of an agile team, was low. In this particular project the lack of participation by the customer in the daily stand-up, iteration planning and iteration retrospectives has had an impact on the relationship with the customer. As a result, the team do not place a great deal of trust in the customer. This suggests that lack of participation by any key team member in these agile practices can result in the development of a lack of trust.

As a consequence of using an agile methodology the team had autonomy to make their own decisions, set their own deadlines for an iteration and could control what they do within the team which were defined during the iteration planning meeting. The project manager listened to the team, did not appear to micromanage and generally supported the decisions made by the team. The environment itself was very supportive and using these agile practices has provided an opportunity for trust to foster and develop among team members. This suggests that it is important for the project manager to allow the team to self-manage, to make their own decisions, and to support them in whatever way possible to help the team to build relationships with each other, resulting in a potential increase in trust, stronger team spirit, and improved performance of the team.

Distributed software development teams face particular challenges, particularly in relation to trust, culture, and communication when the time zones vary dramatically (Ramesh et al. 2006) and it is important to find ways to address these problems.

One possible way is to use agile practices which provide distributed team members with a facility to communicate and interact with the team on a regular basis and help them to feel part of the team. Daily meetings encourage a certain amount of informal communication, such as social conversations as the start of a meeting, which can contribute to breaking down any cultural barriers that may exist and building a relationship with the team members. This may lead to an increase in the levels of trust between team members as current concerns can be raised and discussed, ideas and problems can be shared and advice provided in a constructive way. Feedback is obtained regularly and team members can form their own opinion as to whether other team members are competent and can be trusted to complete good quality work on time. These meetings in addition to the iteration planning and iteration retrospective meetings also provide transparency on tasks and whether or not tasks are being completed on time.

People are extremely important in an agile team and it is imperative that they can work together, have good relationships, and trust each other to deliver on what is promised (Nerur et al. 2005). As the demand for agile software development continues it is important that trust is a core element of a team. Many different agile practices may contribute to trust amongst team members, but the findings of this study are a first step towards understanding how three different agile practices can support and facilitate trust in an agile team. Some of the findings presented here may be of benefit to practitioners when trying to identify the characteristic of individuals or teams who may be suitable for ASD, or where practitioners are attempting to determine the benefit of implementing these agile practices. At the same time it is important to remember that another agile team in a different environment may not use these agile practices in the same way and as a result may not have the same positive experience.

This research is limited by virtue of the fact that a single case study is utilized as the research method. The findings are therefore, only representative of this team. While the team was initially distributed across three continents, team members in one of the locations departed from the team during data collection and were not replaced locally. The perspectives of these team members may have provided different insights into the agile practices used as they were distributed team members of a different culture. A second limitation relates to the number of practices studied. This research only focused on three practices. Future research should examine other agile practices to determine if they also impact on trust amongst agile team members.

References

Abrahamsson P, Conboy K, Wang X (2009) Lots done, more to do: the current state of agile systems development research. Eur J Inf Syst 18(4):281–284

Agile Alliance (2001) Manifesto for agile software development. www.agilemanifesto.org

Alper S, Tjosvold D, Law KS (1998) Interdependence and controversy in group decision making: antecedents to effective self-managing teams. Organ Behav Hum Dec Process 74(1):33–52

Barker JR (1993) Tightening the iron cage: concertive control in self-managing teams. Admin Sci Q 38(3):408–437
Beck K, Andres C (2005) Extreme programming explained: embrace change, 2nd edn. Addison-Wesley, Boston
Benbasat I, Goldstein DK, Mead M (1987) The case research strategy in studies of information systems. MIS Q 11(3):368–386
Bishop JW, Scott KD, Burroughs SM (2000) Support, commitment, and employee outcomes in a team environment. J Manage 26(6):1113–1132
Blomqvist K (1997) The many faces of trust. Scand J Manage 13(3):271–286
Boehm B (2002) Get ready for agile methods, with care. IEEE Comput 35(1):64–69
Cockburn A, Highsmith J (2001) Agile software development: the people factor. IEEE Comput 34(11):131–133
Cohen SG, Chang LEI, Ledford GEJ (1997) A hierarchical construct of self-management leadership and its relationship to quality of work life and perceived work group effectiveness. Person Psychol 50(2):275–308
Conboy K (2009) Agility from first principles: reconstructing the concept of agility in information systems development. Inf Syst Res 20(3):329–354
Cooper DR, Schindler PS (2001) Business research methods. 8th edn. McGraw-Hill, London
Cummings TG (1978) Self-regulating work groups: a socio-technical synthesis. Acad Manage Rev 3(3):625–634
Das TK, Teng B-S (2001) Trust, control, and risk in strategic alliances: an integrated framework. Organ Stud 22(2):251–283
Dybå, T, Dingsøyr T (2008) Empirical studies of agile software development: a systematic review. Inf Softw Technol 50(9–10):833–859
Elssamadisy A (2007) Patterns of agile practice adoption: the technical cluster. C4Media, Santa Ana
Elssamadisy A (2008) Agile adoption patterns: a roadmap to organizational success. Addisson Wesley, Upper Saddle River
Gruenfeld DH, Mannix EA, Williams KY, Neale MA (1996) Group composition and decision making: how member familiarity and information distribution affect process and performance. Organ Behav Hum Dec Process 67:1–15
Hackman JR (1990) Groups that work (and those that don't). Creating conditions for effective teamwork. Jossey-Bass, San Francisco
Hansson C, Dittrich Y, Gustafsson B, Zarnak S (2006) How agile are industrial software development practices? J Syst Softw 79(9):1295–1311
Hofstede G (1980) Motivation, leadership, and organization: do American theories apply abroad? Organ Dyn Summer:42–63
Jurison J (1999) Software project management: the manager's view. Commun AIS 2(17):1–50
Kramer RM (1999) Trust and distrust in organizations: emerging perspectives, enduring questions. Annu J Psychol 50(1):569–598
Lewicki RJ, McAllister DJ, Bies RJ (1998) Trust and distrust: new relationships and realities. Acad Manage Rev 23(3):438–458
Manz CC, Sims HP Jr (1987) Leading workers to lead themselves: the external leadership of self-managing work teams. Admin Sci Q 32(1):106–129
Maruping LM, Venkatesh V, Agarwal R (2009) A control theory perspective on agile methodology use and changing user requirements. Inf Syst Res 20(3):377–399. doi:10.1287/isre.1090.0238
Mayer RC, Davis JH, Schoorman FD (1995) An integrative model of organizational trust. Acad Manage Rev 20(3):709–734
McEvoy J, Butler T (2009) The role of project management in ineffective decision making within Agile software development projects. Eur J Inf Syst 18(4):372–383
McKnight DH, Cummings TG, Chervany NL (1998) Initial trust formation in new organizational relationships. Acad Manage Rev 23(3):473–490
Miles M, Huberman A (1999) Qualitative data analysis. Sage, London

Moe NB, Dingsøyr T, Dybå T (2010) A teamwork model for understanding an agile team: a case study of a Scrum project. Inf Softw Technol 52(5):480–491

Nerur S, Mahapatra R, Mangalara G (2005) Challenges of migrating to agile methodologies. Commun ACM 48(5):72–78

Ramesh B, Cao L, Mohan K (2006) Can distributed software development be agile? Commun ACM 49(10):41–46

Rousseau DM, Sitkin SB, Burt RS, Camerer C (1998) Not so different after all: a cross-discipline view of trust. Acad Manage Rev 23(3):393–404

Schwaber K, Beedle M (2002) Agile software development with Scrum. Prentice-Hall, Englewood Cliffs

Version One (2009) State of agile development survey 2009. http://pm.versionone.com/State of Agile Survey.html. Accessed 31 Mar 2010

Wageman R (1997) Critical success factors for creating superb self-managing teams. Organ Dyn 26(1):49–61

Wageman R, Fisher CM, Hackman JR (2009) Leading teams when the time is right: finding the best moments to act. Organ Dyn 38(3):192–203

Some Reasoning Behind Conceptual Normalisation

Martin Molhanec

Abstract This article deals with some of the basis of the reasoning behind conceptual normalisation and briefly describes how relational and object normalisation can be derived from it. This contribution is only an introduction to this very interesting, serious and infrequently discussed issue. Nor is it a comprehensive paper covering all particular problems and questions and offers precise mathematical proofs of authors' theses presented herein. The idea of this contribution is to create a proper starting point for the discussion of this issue in the frame of an international expert community engaging in conceptual, object and data modelling.

1 Introduction

This article is engaged with the issue of conceptual normalisation. The author has been concerned with this issue for some years but until now the results from his work have been published only at local Czech conferences and seminars [7–14]. So this article is a first attempt to disseminate the author's opinions in the wider scope of the international expert community engaging in conceptual, object and data modelling. The author hopes that his contribution can be a starting point for a wider discussion about these very interesting and serious problems.

2 Motivation and Problem Statement

Designing the right and correct data model is a crucial moment of the whole life cycle of a data-intensive software product or informational system. The most important feature of this model is that there is no redundancy in it. This important feature of the model reflects the lack of redundancy of the real world around us.

M. Molhanec (✉)
Czech Technical University in Prague, Technická 2, 166 27, Praha 6, Czech Republic
e-mail: molhanec@fel.cvut.cz

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_41,

The paradigm of relational normalisation [2, 3] (1st to 3rd normal form and others that are not so common such as BNCF, etc.) is commonly taught in university courses in the field of database theory and design. There are no doubts about the usefulness and advisability of it. Regrettably, there are many opinions about the nonsensicality of the normalisation principle in the field of an object oriented paradigm given that object oriented databases do not use the concept of primary and foreign keys. This is false in the opinion of the author.

It is clear that we do not need the primary and foreign keys for object oriented databases, because normalization is particularly a process of removing redundancy from our data model and the definitions of normal forms for a relation database with the aid of keys are purely the proper formulation of this fact in the relational area. This approach is supported by many experts [1, 5, 16, 17]. But still, there exist some problems:

- There are no definitively and generally accepted definitions of object normal forms.
- Most of the authors have a problem with replacing the concept of relational keys with other correct concept in the definition of object normalisation.
- It is not clear if the same count of normal forms exist in the object area as well as in the relational area.

3 Approach

The approach to this research rose from the following conjecture:

Conjecture: Both the relational and object models are specialisations of more common conceptual model; hence both the relational and object normalisations are specialisations of conceptual normalisation as well.

Here it should be mentioned that one misguided argument which frequently arises in the course of conversation about the necessity of object normalisation and differences between the object and relational approaches.

False Conjecture: The object approach has not and does not need any primary and foreign keys, therefore it is better than the relational approach and its normalisation is thus nonsense.

This argument along with the idea surrounding normalisation as something that relates to the primary and foreign keys implicates an idea that normalisation in the area of an object oriented paradigm is nonsense. However, as previously mentioned, the normalisation paradigm is not about the keys, because the keys in the relational area are only the proper devices that define the normal forms, but normalisation is essentially about redundancy which rises from the absence of redundancy in the real world.

In addition if the definitions of normal forms are reformulated by means of functional dependencies or by means of internal identifiers in the case of the object area the conclusion is more or less the same.

In the following section the possibility of defining the conceptual normalisation is demonstrated and the relational and object normalisation can be easily deduced from it as well. But there is one basic problem to overcome and that is that there are no keys, the same can be said of the object area, with the help of them we would easily realised the definitions of normal forms in the conceptual area.

3.1 Conceptual Model

The first thing we must remember is that the conceptual model has nothing in common with computer technologies, programming or databases. This is, as mentioned above, a model of concepts which occur in the real world around us. So the conceptual model is an outcome of ontological observation of the real world and rises from the ontological model of the universe. Consequently only an ordinary conceptual model will be use without dealing with ontology too much. In this article only the following concepts will be used.

- Object
- Class
- Inheritance
- Attribute
- Composition
- Relationship

Apart from the fact that some theories propose that the conceptual model is identical to the general object model, we cannot anticipate that any characteristics within this conceptual model are related to any specific implementation in computer hardware or object oriented programme language.

Conjecture: There are no unique internal identifiers of the object in contrast to any usual object oriented systems.

This is an elementary assertion. We cannot predicate that any natural world objects have any real internal identifiers. Any unique identification of an object such as a serial number is an artificial property created by a human being for intention of creating a unique identification. But the majority of natural world objects do not have such identification at all. Of course, the unique genetic identification is not taken into consideration. Finally, it is simple to say, that natural object identifiers, modelled by means of conceptual model, do not exist in the real world.

Lemma: There are not any identical classes in light of its intension.

There now follows a brief explanation. Real world classes are created by a different method in contrast to the object oriented programming classes. In typical object oriented programme languages we can create two classes with the same intension, i.e. they will be defined by the same set of attributes and methods and nevertheless they are comprehended as two different classes by the computer system despite the fact that their behaviour and properties will be identical. In the real world such classes do not exist at all (regardless of Plato's concept of ideas). Real world classes are a concept arising from human beings capability to comprehend and perceive the fact that specific sets of objects are similar to each other. Therefore the conceptual model class is defined solely by its extension; the intension is not exactly obtainable. We can only assign different names to the class, but it remains the same class. We have only different synonyms or other names for it.

3.2 The Base for Conceptual Normalisation

All subsequent considerations used in this article arise from simple and basic assumption.

Axiom: There is no redundancy in the real world.

Let's appreciate the following facts. All objects in the real world exist only in one occurrence. Each human being is a unique individual and there is only one occurrence of the ISD 2010 conference etc. In other words, in the real world there are not any two identical instances of the same object existing in the same time and space. The information systems hold a model of the real world by means of included data, so it is clear that as the real world exists without any redundancy, thus the model of that world also has no any redundancy.

Of course, it is not the case of warehouses that use redundancy for the purpose of achieving certain specific features. Similarly, database practises frequently rise above the mentioned principles to achieve an increasing throughput in the database system.

3.3 Definition of Redundancy

Formally we can define redundancy as the non-existence of identical objects in the system. More formally; Let us have two objects x, $y \in \Re$, where $\Re$ denotes the set of all object in the system and *cf* is a function with meaning *return the conceptual meaning of an object on input*, then we assume following definitions:

Definition: A redundant system is a system when the following equation is valid:

$$\exists x, y \in \Re(cf(x) = cf(y))$$

Definition: A non-redundant system is a system when the following equation is valid:

$$\forall x, y \in \Re(cf(x) \neq cf(y))$$

It is assumed that the real world is a non-redundant system; therefore all mappings of this system to any real information system result in a non redundant informational system as well. Simply said the mapping must be *isomorphic* in the domain that is our concern.

3.4 The Conceptual Object and Its Features

The following section introduces the features of conceptual objects commencing with a natural lemma.

Definition: A property is certain a characteristic of an object having a value.

An example of such a property could be: colour or age. Then, the property can also be a set. Moreover, the relationship to other objects can also be a property. Thus, a property is a certain abstraction of an objects characterisation, constituent of its intension and distinguishable or perceivable by a human being.

Lemma: A set of object properties is unique in relation to it.

A car has only one colour property. Of course the colour of car body is a separate property to the colour of the car's chassis. These two colours are two different properties. A person has only one age property, denominated 'age' etc. We can prove the property and the result of this proof is that the property is a concept as well. In other words, a real car does not have two colours and a person does not have two ages at the same time. The following theorems proposed by the author relate to object properties as well.

Theorem: An object property is not dividable, in other words, the object property is atomic. If we need to work with a part of it, the part becoming an object of its own, often abstract, with its own properties.

Theorem: If we need to work with a group of object properties as if it was one concept in itself, the group becoming an object of its own, often abstract, with its own properties.

Theorem: Prospective atomicity of any property depends on domain oriented point of view.

Herein lies an explanation of the above mentioned theorems with the aid of simple examples relating to the name of a person. If we work in the domain of our special-interest, always with the whole personal name as a single property including both first and last personal names, we can comprehend this property as atomic and unique in the frame of the concept of a person.

But, if we need to work, in the domain of our special-interest, with the first and last name separately, then the personal name becomes an abstract concept by itself with two properties (the first name property and the last name property). On the contrary, if the concept of person has two properties, a first and last name, and we need to work with this pair of properties any time jointly, the pair becomes a single concept by itself with its own denomination, in others words, the pair is a new named concept.

Finally, it is important not to overlook the fact that in the conceptual world it is principally possible to think of a property always in a dual manner.

Lemma: A property of any object can be perceived simultaneously as another object related to the aforementioned object.

From this duality and the above mentioned lemmas the result finally provides an explication of the possibility to model the real world in many different ways and all these ways can be right.

3.5 Property, Object and Class

It is possible to define a conceptual class as follows. (It should be noted that we deal with conceptual classes and not with programmer classes.)

Definition: The conceptual class is a named set of properties belonging to each similar object.

We must be aware that in the real world the class is only an abstract concept denoting a set of abstract and real objects which are similar to each other. Thus the class named ‘car’ is only a notation of the set of objects with such similar properties that we have a need to have a common notation for them. This notation is the name of the class. It must be notated that the conceptual class does not exist as a real world object at all. Herein two further definitions are submitted:

Definition: An intension of the class is a set of all possible properties of this class.

Definition: An extension of the class is a set of all possible objects of this class.

In contrast to the world of programmers where it is possible to easily define the intension of a certain class and the extension of this class results from this definition. In the real world only real natural objects exist which constitute the extension of similar objects and the intension is constructed on the basis of the investigation of their similarities.

3.6 Object, Property, Class and Relationship

Up to now relationships have not been dealt with. In agreement with an ontological approach [4] there are four different types of relationships that are recognized within the framework of the conceptual model.

- Relationship between the class and its property.
- Relationship of inheritance between classes.

- Relationship which represents a composition of classes.
- General relationship between objects.

The above mentioned relationships are generally used in different modelling techniques with the exception of the first which is implicitly supposed. It is worth mentioning that although these relationships are defined with one common lexical label, namely ‘relationship’, they are essentially different. All ‘relationships’ are unrelated ontological concepts and common denominations of them can be very misguided.

4 Results—Proposals of Conceptual Normal Forms (CNF)

The following section is a presentation of the definitions of conceptual normal forms which are understood as the rules of redundancy prohibition already introduced above and here altogether mentioned again.

0 *CNF: There is no redundancy in the real world.*
1 *CNF: A set of object properties is unique in relation to it.*
2 *CNF: An object property is not dividable, in other words, the object property is atomic. If we need to work with a part of it, the part becoming an object of its own, often abstract, with its own properties.*
3 *CNF: If we need to work with a group of object properties as if it was one concept in itself, the group becoming an object of its own, often abstract, with its own properties.*

The grounds for these conceptual normal forms have been introduced previously. It is the authors’ belief that the relational and object forms can be derived from these more common conceptual forms. This can be briefly described in the following text.

4.1 Relational Normal Forms (RNF)

The author of this article suggests that 1. RNF can be substantiated by 0. to 2. CNF. The basis for this suggestion comes from the fact that 1. RNF deals with atomicity of data attributes, prohibition of multi-attributes and the necessity of primary key existence. Evidently the issue of atomicity relates to the herein proposed 2. CNF, the prohibition of multi-attributes results from the 1. CNF and the issue of the necessity of the primary key existence relates to principal 0. CNF.

The author suggests that the 2. RNF is a specialisation of a more common 3. RNF, but more detailed discussion of this issue is outside the parameters of this article. This means that both the 2. and 3. RNF follow from the above suggested 3. CNF. The evidence is based on the consideration that transitive dependency between relational keys at the level of the relation data paradigm is simply an implication of the incorrect recognition of the conceptual object at a higher level of comprehensibility.

It is worth noting that the concept of relational keys in relational database systems by itself presents only the programmer implementation of the concept of functional dependency by the implication of mutual relationships among conceptual objects. Thus, the incorrect recognition of an object at the conceptual level leads to transitive dependency between relational keys in the relational level.

4.2 Object Normal Forms (ONF)

At this time there is not any standard or commonly accepted concept of object normal forms, notwithstanding there are many scientists engaged in this issue. A brief summary of these is included in [6, 15]. This work contains definitions of object normal forms based on an approach introduce originally by Ambler in [1] and further elaborated in the aforementioned work by Merunka and Molhanec.

Despite the fact that the author deal with the object oriented paradigm, the definitions of object normal forms are simply based on analogical relational forms, i e., 1. ONF is based on 1. RNF, and so on. The fundamental difference lies in the fact, that the definitions of all ONF are constructed without the use of the relational keys of course. Thus, the presented definitions of ONF in the aforementioned work are very similar to the definitions of CNF proposed herein and we can apply the same or very similar argumentation for their reasoning.

5 Conclusion

The author of this article suggests that relational and object normalisation arises from the same source i.e. from the conceptual normalisation. Surprisingly, this principal and serious subject matter is not widely discussed in the expert community at all. In addition, the need for object normalisation that directly arises from the conceptual normalisation is often disputed.

The author believes that his contribution in this very interesting and serious subject matter can be a good starting point for the discussion of this issue in the frame of the international expert community engaging in conceptual, object and data modelling.

Acknowledgements This research (work) has been supported by Ministry of Education, Youth and Sports of Czech Republic under research program MSM6840770017, SGS10/267/OHK3/3T/13 and SGS10/162/OHK3/2T/13.

References

1. Ambler S (1997) Building object applications that work, your step-by-step handbook for developing robust systems using object technology. Cambridge University Press, Cambridge. ISBN: 0521-64826-2

2. Christopher JD (2011) Wikipedia. http://en.wikipedia.org/wiki/Christopher_J._Date
3. Edgar FC (2011) Wikipedia. http://en.wikipedia.org/wiki/Edgar_F._Codd
4. Heller B, Herre H (2004) Ontological categories in GOL. Axiomathes 14(1):57–76. Kluwer
5. Khodorkovsky VV (2002) On normalization of relations in databases. Program Comput Soft 28(1):41–52. January 2002, Nauka Interperiodica
6. Merunka V (2006) Datové modelovàní. ALFA Publishing, Praha. ISBN: 80-86851-54-0
7. Molhanec M (2002) UML–několik kritických poznàmek. Sborník konference Tvorba software 2002. Tanger, Ostrava, pp 150–159. ISBN: 80-85988-74-7
8. Molhanec M (2003) Objektové metodologie–jejich užití a výklad. Sborník konference Tvorba software 2003. Tanger, Ostrava, pp 111–115. ISBN: 80-85988-83-6
9. Molhanec M (2004) Několik poznàmek k porozumění objektového paradigmatu. Sborník konference Objekty 2004, ČZU PEF, Praha, pp 189–197. ISBN: 80-248-0672-X
10. Molhanec M (2004) Kritika některých výkladu objektově orientovaného paradigmatu. Sborník konference Tvorba software 2004. Tanger, Ostrava, pp 163–172. ISBN 80-85988-96-8
11. Molhanec M (2005) Konceptuàlní modelovàní. In Objekty. VŠ Technickà Universita, Ostrava. ISBN: 80-248-0595-2
12. Molhanec M (2005) Zàsady konceptuàlního totàlně objektově orientovaného modelovàní. In: Tvorba softwaru. VŠB, Ostrava, pp 153–158. ISBN: 80-86840-14-X
13. Molhanec M (2006) Konceptuàlní modelovàní, formàlní zàklady a ontologie, In: Tvorba softwaru 2006. VŠB – Technickà univerzita Ostrava, Ostrava–Poruba, pp 121–127. ISBN: 80-248-1082-4
14. Molhanec M (2006) Ontologické zàklady konceptuàlní normalizace, In: Objekty 2006. Technickà universita Ostrava– Vysokà š kola bàňskà, Ostrava, pp 81–90
15. Molhanec M, Merunka V (2007) Object normalization as the contribution to the area of formal methods of object-oriented database design. In: Sobh T (ed) Advances in computer and information sciences and engineering. Springer, Berlin, pp 100–104. ISBN: 978-1-4020-8740-0
16. Tari Z, Stokes J, Spaccapietra S (1997) Object normal forms and dependency constraints for object-oriented schemata. ACM Trans Data Syst 22(4):513–569 (December 1997)
17. Yonghui W, Zhou A (2001) Research on normalization design for complex object schemes. Info-Tech and Info-Net 5:101–106. Proceedings of ICII, Beijing

Improving Component-Based WebGIS Development by Adopting a Software Product Line Approach

Juan Manuel Moreno-Rivera and Elena Navarro

Abstract Development of Web-based Geographical Information Systems (WebGIS) is mainly done by Small Enterprises which usually employ Open Source Software (OSS) components. The number and complexity of requirements for these Information Systems has exploded during the last years due to the technological advances. In addition, enterprises working on WebGIS must carry out intensive software development within short time frames. This situation is claiming for an improvement of their software development processes so that both the time to market and costs can be reduced. In this paper, we present the situation of a research-in-progress performed by one of these SMEs working on WebGIS. Software Product Line approach has emerged as a promising solution to face the problems due to the situation in the WebGIS domain. We have also detected that modeling tools can help WebGIS developers to reduce the time to market and cost of their developments and, at the same time, offer a quality warranty and more robust ISs. We have started a Software Process Improvement and its first steps have been taken whose results are presented here.

1 Introduction

The development of Information Systems (IS) is becoming more complex every day. There are more data available in the World Wide Web and the new technologies are pushing towards a new generation of IS. Most of the enterprises working on Web-based IS are Small and Medium Enterprises (SME) [1]. WebGIS domain is not an exception to this situation. The complexity of these systems has exploded and has produced a situation in which the development of new products is becoming harder

J. M. Moreno-Rivera · E. Navarro (✉)
SIGTel Geomàtica, Centro I+D de Empresas and Grupo de Teledetección y SIG, Instituto de Desarrollo Regional, University of Castilla-La Mancha, Albacete, Spain
e-mail: JuanManuel.Moreno@sigtel.es

E. Navarro
Computing Systems Department, University of Castilla-La Mancha, Albacete, Spain
e-mail: Elena.Navarro@uclm.es

J. Pokorny et al. (eds.), *Information Systems Development,*
DOI 10.1007/978-1-4419-9790-6_42,

whereas their time to market is being reduced in order to be more competitive. In [2], it is detailed how short time frames for Web-based projects is affecting their process development and final products, which also affects to WebGIS projects. Unfortunately, more often developers give up quality in order to accomplish deadlines, redo work devote no time to documentation, etc. In order to solve these kinds of problems and to improve the development process within our enterprise, we start our research in 2009. Our idea is to apply the Software Product Line (SPL) approach [3] to help us to improve our production chain and the maintenance of new products as well as the components and libraries to be reused in future works.

In this work we present our preliminary outputs of a research that we started within a typical SME from the WebGIS domain, SIGTel [4]. This work has been structured as follows. In Sect. 2, we present the context in which our research is being carried out. In Sect. 3, we analyze the state of art and problems found in WebGIS development. In Sect. 4, the components that make up these IS, most of them Open Source Software (OSS) components, are analyzed to highlight to what extent we can exploit reusable assets and opportunities automate the development process. Then, the SPL approach selected to be implemented within our enterprise is introduced in Sect. 5. The Software Process Improvement (SPI) that we are carrying out is defined in detail in Sect. 6. Finally, conclusions and future work is presented in Sect. 7 as well as the initial outputs of this research.

2 Research Context and Methodology

SIGTel is a Spin-Off whose founders have been the Earth Observation and GIS group (EOGIS) [5] of the University of Castilla-La Mancha (UCLM) [6]. The first author has been involved in WebGIS developments based on OSS technologies since 2003. His work has been mainly carried out in the context of two European Projects in EOGIS: DEMETER [7] and PLEIADeS [8]. The high number of WebGIS developments carried out by EOGIS during the last five years so that the group has required EOGIS to establish a technological base and personal resources for the development of WebGIS applications.

The synergy of SIGTel and EOGIS (see Fig. 1a) is based on the EOGIS role as a R + D + i department, being in charge of establishing new research lines and adapting the existing ones to new market requests. SIGTel is responsible for including new products and services in its portfolio by considering both existing and new developments of the EOGIS group. SIGTel also carries out market analysis tasks to provide EOGIS group with feedback of market status and new trends. Then, the EOGIS group works according to the feedback received from SIGTEL updating and customizing their components to new demands as well as adding new features based on research projects. This synergy provides research results that can be directly applied to real life solutions. However, the SIGTel needs of generating market benefits from the results of the EOGIS group is not an easy task taking into account it has software process in which reuse and generation of new products are not as efficient as they should be.

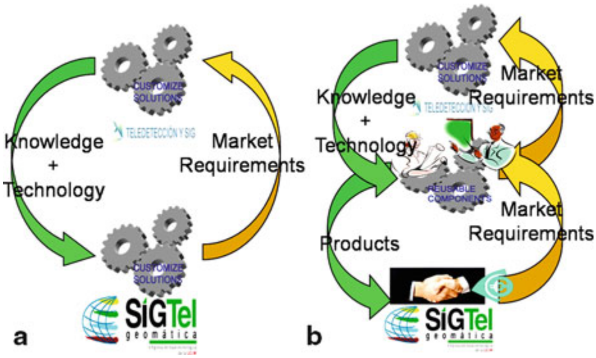


Fig. 1 Synergy EOGIS—SIGTel: **a** present situation, **b** expected situation

In this context, skills requirements of developers are higher than ever and a continuous learning process is necessary to keep track of new technology emerging every day. The short time frames associated to projects developed within this domain, are forcing SIGTel to reduce time to market while dealing with long periods of adaptation and refactoring to include EOGIS new components into the production chain. This situation claims for a SPI in order to reduce the problems found in the technology transfer between these two groups and, to ensure that the process does not sacrifice products' quality and client satisfaction for the sake of accomplishing scheduled times. Due to this situation, we are working for improving the synergy between the two groups looking for the state shown in Fig. 1b. We have to consider the challenges found in WebGIS domain and the ones related to deploy methodologies in SME [9].

The first step we have done is to evaluate alternatives to deploy a systematic process and to determine which ones of the existing core assets should be firstly adapted according to their potential benefits. Identifying the critical elements to be improved within the actual EOGIS-SIGTel synergy is another important challenge we have faced. As SIGTel has decided to make research into SPL as the best option to select the adequate methodology, our final objective is to deploy a solution based on this approach that can improve the final products quality. Finally, we have selected IDEAL [10] as the methodology to conduct this SPI. Our initial results are presented in Sect. 6.

3 Web-Based Geographical Information Systems (WebGIS)

A GIS is any IS that captures, stores, analyzes, manages, and presents data that are linked to location. The Web-based version of these systems is becoming more and more popular every day. Nowadays, everybody knows or uses a WebGIS, although he/she even does not know it. The most popular example is Google Maps [11], but there are other options like Yahoo Maps [12] or Bing Maps [13]. These examples have got a very basic functionality but they are very familiar to internet users. WebGIS main feature is a map, the center of the system, along with functionality to analyze, edit and produce spatial information.

3.1 State of the Art

In the last decade, GIS has jumped off from the desktop to the web, passing from being very heavy process applications for users with a high technological level, to applications for all kind of internet users. This situation has increased the demand of these systems and has meant a change of the market. Demand has exploded and a lot of small and big enterprises are fighting for a place in this emerging market. New users are becoming more and more familiar with WebGIS and are requesting new features. Experienced users, coming from desktop GIS, are requesting also new features in addition to those functionalities already offered by desktop versions. Heavy desktop processes are being adapted to the web as soon as the network and technologies are improving and supporting the change. Due to the increasing complexity of WebGIS, software development enterprises are expected to develop new WebGIS systems in less time, with more and better features and increasing quality. This situation is becoming very exhausting and reuse is becoming a hard task. Each component needs to be adapted and be more flexible for new changes as long as market requests new features. For SME, this situation is even more critical and SIGTel is not free of these problems.

Besides, it is very hard to maintain the production level and still being up to date of new market trends. Big enterprises have a R + D department with many resources devoted to have its technology up to date. But a SME that develops WebGIS following a project-based process, cannot attain this task with its traditional software development methodologies. The time to market of final products is too long and their quality is not what users expect also due to the difficult tasks of requirement engineering and testing WebGIS systems. The question is that these problems should not happen because the process of developing a new WebGIS is usually based on reuse of existing components and in many cases consist of the adaptation of previous systems. There are also communications standards defined by the Open Geospatial Consortium (OGC) [14], widely used and accepted by the community, that describe how the communication between components can be carried out through web services in most cases. These standards promote the component-based development and a strong data interchange between different WebGIS located all around the world. This network of WebGIS is complicating the development of new WebGIS at the same time that opens market opportunities for enterprises and new research trends, like business intelligence or the cloud as a new platform for offering services based on WebGIS.

4 OSS Component-Based Development in WebGIS

The maind reasons behind the rise of OSS components available for WebGIS development are open standards and an active community. OGC has defined standards that cover almost all the needs of these systems. One of the advantages of this organization is that is strongly linked to market needs and reacts according to its needs. For this reason, the standards defined by OGC are widely used and easier to use than others because of their simplicity. These standards support communication between different WebGIS as well as different components within the same WebGIS.

The main OGC standard is the Web Mapping Service (WMS),which defines the way maps should be requested from a mapping client. Recently, the OGC has released the WMS Tiles (WMST) standard which originates from the demands of the community to reduce response time for the main activity within WebGIS, maps generation. The Web Feature Service (WFS) is in charge of defining the way vector information is requested and transferred. The Web Coverage Service (WCS) supports interchange of raster data. Another important standard is Web Processing Service (WPS) that supports the communication for processes request. Finally, the emerging Sensor Web Enablement (SWE) family of services provides support for real time sensor data interchange to simulate environmental models and monitor emergencies. This family of standards covers almost all data interactions between these IS.

Based on these communication standards, a huge number of components has emerged and that provide WebGIS developers with different functionalities included in a WebGIS. These components are commonly used in two different ways, as source code or for execution support. However, although the first option is widely used it complicates even more the interconnection and reuse of the different components. The growing number of available components has originated a difficult situation with a lot of new challenges that the developing groups should face. Bellow those most significant are listed:

- *Non standardized documentation:* the different components, frameworks and pieces of reusable software do not follow a standardized documentation process. As it is an open and free community, the different project leaders define their own minimum design and development documentation if they do it, because most of these projects are barely documented.
- *Missing quality warranties:* there is no information about the quality of the available components. An enterprise should intensively test each one of these components as well as test their connection to those existing ones in their own systems. However, it is quite difficult considering the complexity of a WebGIS.
- *OSS projects live time is never known:* another important benefit of OSS is that there is a community supporting these pieces of software. However, in OSS you never know how long this community will support them. So, your team needs to be able to complete or supply this support in case your own components are not good enough.

There are initiatives that can be used to support the development of these systems. OSGeo [15] is an example that originated in the Open GIS community as a platform to support a standardized and warranted family of OSS projects covering almost all aspects involved in a WebGIS development. Although, it has shown to be very useful, not all projects can assign the requirements minimum to be part of this family. It has motivated that other projects outside OSGeo are gaining the attention of WebGIS developers. This has motivated that software development projects instead of being agile and with a short time frame are now too complicated and with too time behind the schedule, that is, providing enterprises with no profit. Developers must always

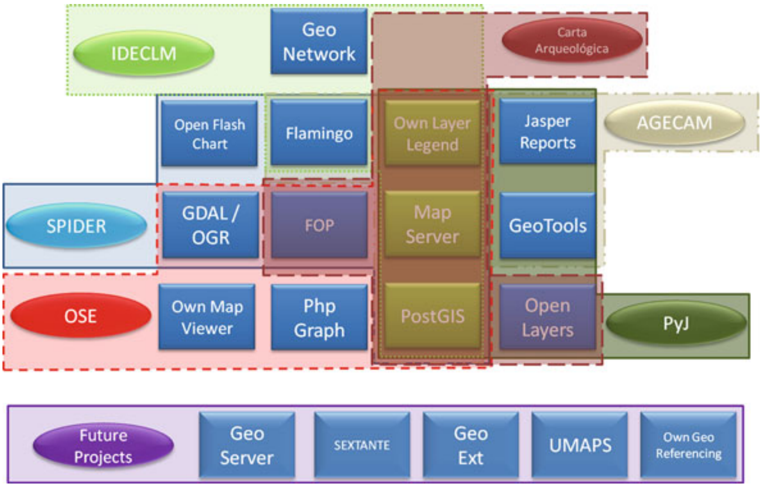


Fig. 2 Components involved in different OSSS WebGIS developments from SIGTel

work with documentation, when available, prepared in many different ways. It is also necessary to develop systems with the flexibility necessary to support changes of either user requirements or technology launches. Finally, SIGTel needs to get more involved in OSS community to follow the evolution of the third party components and be always dealing with the last technologies. Most of SME enterprises do not accomplish an efficient interaction with OSS communities. This situation is very usual due again to the timeframes and resources constrains that SME suffer.

In order to understand to what extent OSS components have been adopted for WebGIS development, and as part of the domain analysis process we are carrying out, Fig. 2 shows a sample of the most significant WebGIS systems produced by SIGTel. The objective is to analyze which are the core components that every WebGIS includes are, which are optional and which are necessary. This figure provides a general overview of the elements involved in a WebGIS development and its complexity.

We can easily draw two essential conclusions from the figure. First, there is a high interaction level between own and third party components. For example, one of the core components common to all projects (PyJ project in this figure) has been developed by us and also there are API's for supporting the compatibility for the different viewers, own and third party ones. This means an intensive and continuous learning process in order to follow the evolution of third party components and a hard effort to accomplish compatibility with our own components. Second, there are some technological aspects in SIGTel that are supported by only one component. It represents a comfortable situation because they are components well-known by SIGTel developers. However, it can become a hazardous situation because changes can cause a hard migration. Therefore, this situation requires us to strike a balance between maintaining our own components, and using third party components that should be updated and migrated, and integrated in an efficient software process development. Also, the figure tries to represent the next generation of own and third party components to be included as part of the technology included in our portfolio. They are examples of new components that are going to complicate the architecture of our systems. If we have problems now, the situation will be worst when these components are incorporated.

5 Software Product Lines Approach

SPL has emerged as an alternative in SIGTel to establish a systematic and productive process for reusing existing software components and other assets shared by the EOGIS group. SPL approaches also provide an ideal platform to define the appropriate framework to support the technological transfer between EOGIS and SIGTel in an efficient and productive way. One of the reasons to propose a SPL approach for SIGTel is that SPL concepts are very familiar to EOGIS and SIGTel developers. There have been previous SPI attempts to establish basic components and a shared architecture that could be reused in different projects. The first attempt was the definition of reference architecture for WebGIS. It was used to provide a solution to three different clients. After its development, a technological change brought out this reference architecture was out of date because its design was strongly coupled to the technology. The second attempt was to develop a core map viewer component for WebGIS that could be included in every related project. It was used widely to produce cloned versions of this component for other projects, but the maintenance of the cloned systems increased the costs so that this component felt into disuse. These examples show how a similar approach to a SPL has been already used in SIGTel and EOGIS groups.

From the existing SPL approaches, we have selected a component-based approach, such as [16] and [17], for the development of the SIGTel SPL is. As was stated in Sect. 4, WebGIS development usually focuses on combining existing components. Therefore, this approach that exploits components as basic units to work on seems ideal for the domain situation. The basic idea of this approach is the integration of different components by using their own public interface and, each component is also internally configured according to user specific requirements. In WebGIS domain, the role of this public interface can be played by a mixture of API's and OGC-based web services. For example, a combination of a map viewer and a legend layer component could be based on an API of the viewer that is used by the layer legend. On the other side, the map viewer would be connected to the map server through OGC standards to request the data to be shown. An example of customizable components could be a system that requires spatial data edition. The map viewer could be configured to offer this functionality to the user. Otherwise, this functionality could be removed of the component in order to avoid complicating the interface with unnecessary tools.

Unfortunately, this is how components usually interact within a WebGIS and the way decisions are taken during their development. Therefore, implementing a SPL approach, whose reference architecture can be configured by adding customized versions of components (see Fig. 3), seems ideal to supply an automated process to

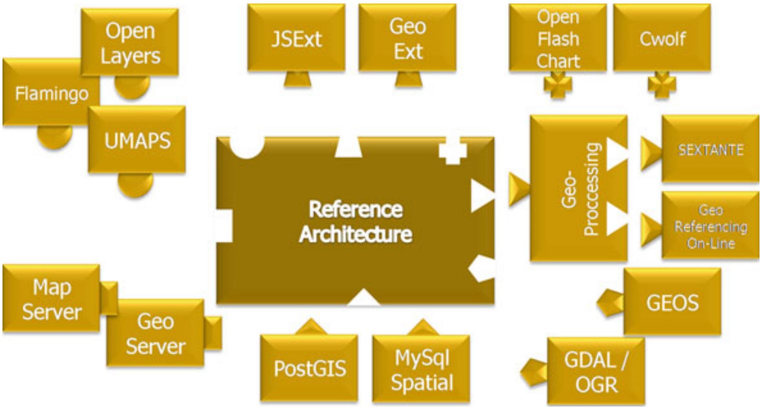


Fig. 3 Present and near future components to be included in SPL core components

develop new systems. In addition, by offering a modeling tool to design new systems would help to increase the number of prototypes easily generated according to initial user requirements. This would solve the problem of most WebGIS users who usually do not know the potential benefits of these tools. The lack of knowledge about these systems causes a continuous updating process that could be avoided thanks to the fast prototyping option supported by a SPL approach.

Another expected benefit of applying this kind of approach is that when generating new systems, documentation inherited from different components could be generated for the new system in an automatic way. This means that we would have the opportunity of carrying out an agile approach without giving up documentation quality. In addition, test benchmarks of new systems could start using group of benchmarks inherited from the different components included in the system. Finally, an integrated environment that allow us to model new systems and implement specific requirements, would give our team developers a perfect tool to support the process of implementing new WebGIS based on an efficient and extremely productive methodology.

6 Software Process Improvement in SIGTel and EOGIS Group

The methodology chosen to carry out our SPI has been a customized version of IDEAL [10]. IDEAL, as can be observed in Fig. 4, entails five phases: *initialing*, *diagnosing*, *establishing*, *acting*, and *Learning*. A brief introduction to the actions planned for each phase according to the instructions from [10] is shown in Table 1.

The stimulus for change was originated in an internal evaluation process of SIGTel that concluded the need for reuse improvements within the project developed by the enterprise. Another conclusion was that there is a problem of technology transfer between EOGIS group and SIGTel. Due to these conclusions, meetings with senior

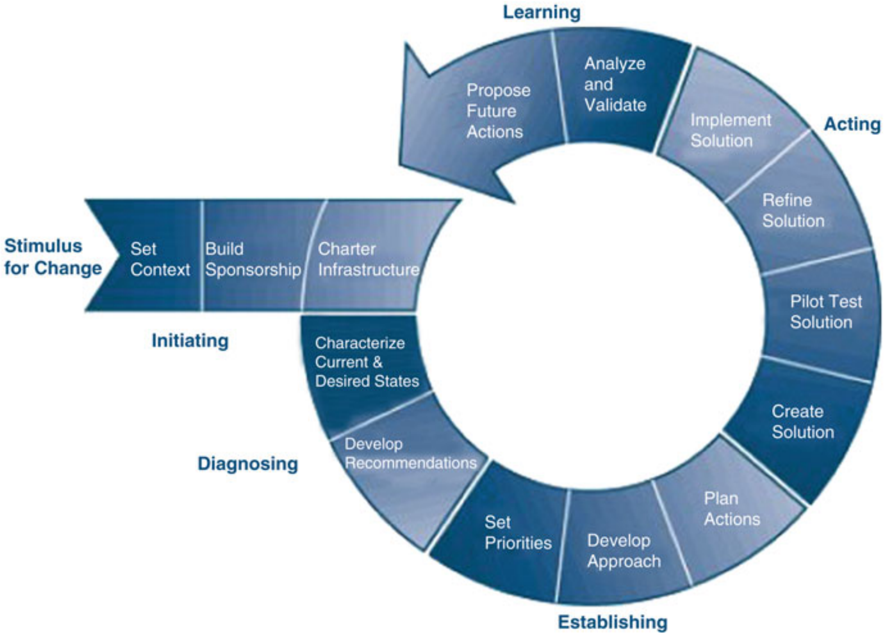


Fig. 4 The IDEAL Model. [18]

Table 1 IDEAL detailed tasks for the SPI in SIGTel

Phase	Tasks to be developed
Initiating	To define roles To schedule meetings and present the SPI plan to the rest of involved people
Diagnosing	To gather and analyze previous SPI efforts To evaluate the present situation in SIGTel according to FEF To identify critical problems in the current process during different scheduled meetings To coordinate with the new SPI initiative other specific SPI tasks already proposed
Establishing	To define specific goals for the first iteration for each of the four dimensions according to the FEF model To choose, if necessary, extra metrics apart from FEF to evaluate state evolution
Acting	To develop a prototype based on ECLIPSE To test prototype within developers core group To refine solution based on feedback from testing group To present proposed solutions
Learning	To evaluate performance of the roles and groups To perform a final evaluation according to FEF model Is that solution enough or is necessary another iteration?

developers and project chiefs were arranged and some roles were defined to support and coordinate this process during the initiating phase. Three projects were selected to devote part of their resources to test reusability of core components and report problems and suggestions about how to adapt these core components to future projects' requirements. Then, during the diagnosing phase, previous SPI efforts to improve reusability were analyzed and an evaluation of the existing situation was carried out based on FEF [19]. Based on the results, the critical problems were identified. To finish this phase the chosen projects and existing initiatives for improvement within the enterprise were coordinated. Nowadays, we are finishing the establishing phase by defining the specific goals for this first iteration and looking for extra metrics, if necessary, for measure changes.

After finishing the establishing phase, a prototype of the modeling tool to support the SPL approach is going to be implemented as the first step from the acting phase. This tool is going to be based on the ECLIPSE Framework [20], as it is already being used in both environments, SIGTel and EOGIS. So, it would represent a less aggressive change than supplying a framework totally new that would compel changes in their basic developments tasks. The development of this tools is going to be accomplished based on the task shown in Table 1. Finally, after the action phase, during the learning phase we are going to evaluate results and produce input for the next iteration.

There are two main risks to accomplishing this SPI transition. The first is related to budget. Due to the existing project-based philosophy, with strong time frames constrains, we have a reduced budget to carry out this SPI. This could cause problems to fund future iterations and, thus, different project managers should get an agreement about this issue as they potentially could benefit from this SPI improvement. Another issue is related to define the SPI in a way that developers feel as part of the change. Thanks to the meetings already arranged, all the developers agreed that a change is necessary as they have realized how necessary it is. However, this situation should not be exploited to impose decisions because all changes to be adopted should be taken according to their needs. In a small enterprise, developers' motivation is a critical factor for a successful SPI.

7 Conclusions and Future Work

Currently, the SPI establishing phase of the first iteration has almost been finished. The changing process has already started in the form of periodical meetings between SPI roles and all developers, as well as other roles within SIGTel and EOGIS group. Actions to be taken in the near future are being planned. As we are a SME enterprise, developers should notice that they are part of change; otherwise it would cause lack of motivation and a negative attitude towards this change. In addition, communication is easy due to the low number of developers and people involved, but it also means that we do not have a huge budget dedicated to fund the implementation of the SPL approach. It makes necessary to encourage developers for this change.

The role of the OSS community is the one corresponding to a complicated and continuously changing supply chain for the enterprise product family. It is necessary to cover all the strategic dimensions present in typical development strategies [21]. The adoption of a SPL approach could be a successful change that could increase productivity in this context. Reducing time to market will first make SIGTel products more competitive. Finally, reducing the number of resources dedicated to produce new systems, would free others resources that could be dedicated to R + D + i tasks, helping on the hard and exhausting task of keeping up to date group's technology. This first iteration is being focused on research tasks, so that after implementing the prototype, an evaluation will be carried out in all the enterprise so that the final deployment plan will be defined in detail.

The results of this process have already started to spring off in the form of a task dedicated to improve basic development practices within the enterprise. Actions to improve basic development tasks have already started to be applied as part of this first iteration. The expected result, as mentioned above, is to improve the reuse of components. In order to check if our objectives are being accomplished, another evaluation according to FEF is going to be the first task to be done in the learning phase. Based on the results generated during this first iteration, different concepts that can be found in a SPL approach will be evaluated and it will be decided which of them, if not all, are included in the final SPI plan.

Acknowledgements This work has been partially supported by SIGTel, EOGIS from the University of Castilla-La Mancha and the Ministry of Science and Innovation from Spain through the Torres Quevedo Program. Special thanks to the team of EOGIS and SIGTel, who provided the necessary information to identify the main core assets, presented in Sects. 4 and 5.

References

1. EC SME Definition – Small and medium sized enterprises (SME) – Enterprise and Industry. http://ec.europa.eu/enterprise/policies/sme/facts-figures-analysis/sme-definition
2. Lang M (2009) The influence of short project timeframes on web development practices. Information Systems Development, Nanchang, pp 1–13
3. Pohl K, Böckle G, Linden FJVD (2005) Software product line engineering: foundations, principles and techniques. Springer, New York
4. SIGTel Geomàtica. http://www.sigtel.es
5. Teledetección y SIG Teledetección y SIG – Inicio. http://www.teledeteccionysig.com/
6. Universidad de Castilla-La Mancha. http://www.uclm.es/
7. DEMETER partners DEMETER. http://demeter-ec.net/
8. PLEIADeS Consortium PLEIADeS Workspace—Home. http://www.pleiades.es/
9. Päivärinta T, Larsen EÅ, Ingulfsvann L, Kristiansen ⊘ (2009) Method deployment in small- and medium-sized firms: addressing the organizational context. Information Systems Development, pp 65–77
10. McFeeley B (1996) IDEAL: A user's guide for software process improvement. Software Engineering Institute, Pittsburgh
11. Google Google Maps. http://maps.google.es/
12. Yahoo Yahoo! Maps. http://maps.yahoo.com/
13. Microsoft Bing Maps. http://www.bing.com/maps/

14. OGC Consortium: OpenGIS® Standards and Specifications | OGC®, http://www.opengeospatial.org/standards
15. OSGeo OSGeo.org | Your Open Source Compass. http://www.osgeo.org/
16. Atkinson C, Bayer J, Muthig D (2000) Component-based product line development: the KobrA approach. 1st Conference on SW Product Lines, pp 289–309
17. Atkinson C, Bayer J, Muthig D (2002) Component-based product line engineering with UML. Addison-Wesley, New York
18. IDEAL Model (2009) Carnegie Mellon University
19. Linden FJVD, Schmid K, Rommes E (2007) Software product lines in action: the best industrial practice in product line engineering. Springer, New York
20. Eclipse Framework. http://www.eclipse.org/
21. Kruithof GH, Meijler TD (2009) Strategic dimensions of software development in a supply chain. Information Systems Development, pp 1–9

The Adaptation of a Web Information System: A Perspective of Organizations

Aivars Niedritis and Laila Niedrite

Abstract We provide a different view on the problem of Web Information System (WIS) adaptation, looking from perspective of organizations that are interested in an adapted Web Information System for their needs if a unified system to support similar business processes is used. We propose an adaptation architecture for WIS. Two levels of adaptation are introduced—coarse grained adaptation for the organization level and fine grained adaptation for the user level. The architecture supports also the situation, when users can work with many instances of the system adapted for different organizations, which are integrated into one instance for a particular user.

1 Introduction

World Wide Web is not only an information source but it also has become a platform for application development. Fraternali [1] described a Web application as a hybrid between a hypermedia and an information system. He stated requirements for Web applications [1] and one of them was: the customization and possibly dynamic adaptation of content structure, navigation primitives, and presentation styles.

According to this requirement the data, navigation, and layout should be considered separately if an adaptation is performed. In [2] it is defined that a Web-based Information System (WIS) is adaptive if it is able to modify and personalize delivery of contents and services according to the context of the client. Personalization requires the adaptation of the applications as much as possible according to the preferences of the user and the context of the user [3].

In [3] the context is considered as a set of properties that describe the environment, where the user interacts with a WIS, e.g. in [3] the context is defined by the time, place, device, user, and environment dimensions.

A. Niedritis · L. Niedrite (✉)
University of Latvia, Riga, Latvia
e-mail: Aivars.Niedritis@lu.lv

L. Niedrite
e-mail: Laila.Niedrite@lu.lv

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_43,

A profile is a representation of an autonomous aspect of the WIS usage context [2]. Another term used in adaptation sphere is the definition of configuration as a specification how information has to be delivered to the user [2].

Many architectures and approaches [2–7] exist for the WIS adaptation, based on different understandings of context, profile, and configuration. Also different properties of each notion are considered.

The authors of [3] provide a meta model, where a profile and context are defined, and they provide also a set of services for the definition of a personalized access model (PAM). PAM definition allows performing different personalization scenarios, including, preference-based recommendation, context-aware content delivery, personalized access to multiple contents [3].

In [2] an approach to the adaptation of a Web information system according to the context is presented. The authors use a general notion of profile, which is associated with a configuration. The configuration specifies how information has to be delivered according to the requirements of adaptation for the profile.

The paper [6] presents architecture for the personalized presentation layer of WIS to support navigation and different views on the presented data.

At the same time to deliver to the user the appropriate WIS also the approaches to the development of flexible and cost effective systems should be considered.

Software product line approach (SPL) [8] is based on predefined architecture and well known core functions. All reusable components are planned in advance. To get different instances of a particular software in traditional architectures usually the application code is changed. The SPL treats variations in different ways [9]: e.g. inclusion or exclusion of elements, usage of different versions of each component. Different can be e.g. behavior and quality features. For the SPL approach the architecture and configuration management is essential, because each new software product consists of numerous core components and variations.

Software as a service (SaaS) [10] defined as a "software deployed as a hosted service and accessed over the Internet" actually introduces a new way of providing the access to the software. SaaS can be services provided to organizations of different size to support business processes common for these organizations e.g. CRM applications. The organizations subscribe to use the software and pay for the usage. The service vendor hosts the SaaS application on their servers and maintains the software and infrastructure. This is often a cost efficient solution for small and medium size organizations.

SaaS applications can be provided at four different maturity levels, starting from serving a customized application instance per each customer on the SaaS vendor's server. More advanced way is to use configurable, multi-tenant single instance solution. In this case the customization is performed using the metadata based configuration of SaaS application to adapt the single common application instance to different customer needs. The multi-tenancy means that users from different organizations use the same application instance, but data are distinguished between tenants of the service.

To effectively solve the problems concerning scalability, configuration and multi-tenancy, the SaaS applications need architectures designed for these purposes.

These architectures also should resolve problems concerning the multi-tenant customization.

Our architecture is provided for the context of WIS usage, where many similar organizations use the same WIS, but each of them gets an adapted instance of the system. The two level adaptation is provided. Also the users get their own adapted instance of the WIS.

The article is structured as follows: Sect. 2 presents our proposed adaptation architecture of a Web Information System and describes two levels of adaptation. We provide description of a case study of the usage of proposed architecture in Sect. 3. In Sect. 4 we discuss our implementation results and make conclusions.

2 Architecture

We propose the adaptation architecture of a Web Information System (WIS) that supports the usage of WIS in many organizations of the same or connected business areas with similar processes. The architecture adapts the initial configuration of this WIS for each particular organization considering their specific needs.

The adaptation architecture of a Web Information System (Fig. 1) consists of the following components:

- Context monitor identifies the context properties of the system usage in the time, when the user connects to the WIS,
- Usage monitor is planned to collect data about the usage patterns of the adapted WIS instances. The goal of this action is to provide information for the monitoring and analysis of processes supported by the WIS and to use these data to develop a strategy for the successful further development of adapted WIS.
- WIS data layer contains profiles that are used in both adaptation levels and the business data. During the coarse grained adaptation process a virtual data store is build along with the organization level WIS instance. The virtual data store contains the data owned by the organization. On the picture (Fig. 1) the virtual data stores for different organizations are depicted as "B_SYS1 Business data" etc. WIS data layer also stores the data about the usage of the adapted WIS instances, collected by the usage monitor.
- Adaptation component starts with the identification of the initial configuration of WIS that is defined according to the proposed architecture.

The adaptation component performs the adaptation in two levels:

- Coarse level adaptation process recognizes the groups of functions that are accessible to the system user according to the usage context, e.g. access point, time, and organization. According to the profiles defined in the system, e.g. configuration profile, this process adapts the WIS initial configuration and establishes an organization level instance of WIS. On the picture (Fig. 1) the function groups are depicted as FG1, FG2 etc. The organization level instances of WIS are depicted as B_SYS1, B_SYS2;

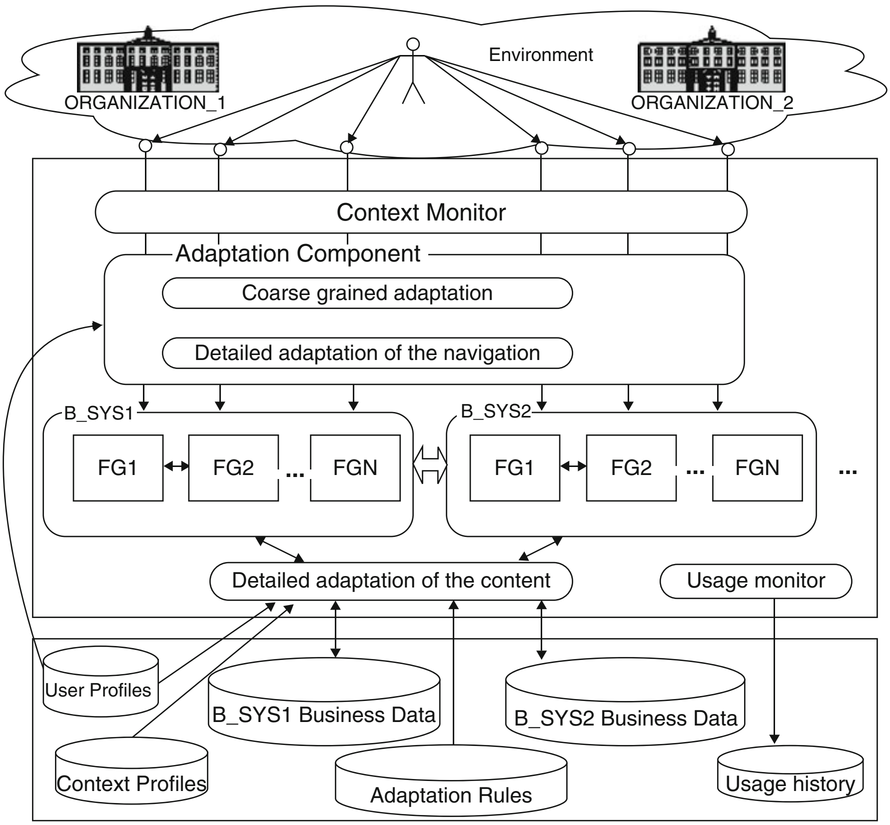


Fig. 1 Adaptation architecture of WIS

- Detailed level adaptation is based on the information stored in the user profile. The detailed adaptation consists of two steps:
  – Detailed adaptation of the navigation uses the profile information about the functions accessible to the particular user and within the framework of the organization level instance of WIS, provides an adapted navigation between all user functions.
  – Detailed adaptation of the content constructs an instance of WIS for the particular user according to the user profile, which describes the data accessible to the user, and to the user functions identified previously.

These components will be described in the next sections of this paper in details.

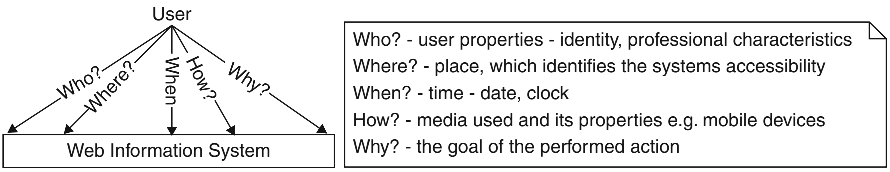


Fig. 2 The usage context of the system

2.1 Context Monitor

The usage context of the system is characterized by properties of the environment, which can influence the interaction between the user and the system in the time moment, when the interaction happens. The usage context is defined by the user, place, time, media used for the interaction, and the intention of the user to perform a particular action by means of the system (Fig. 2).

The goal of the context monitor is to recognize the properties of the environment and to forward them to the adaptation component for usage in the adaptation of the WIS according to the particular environment.

2.2 Definitions

The profiles used in the proposed architecture will be defined by the UML class diagrams. Each class diagram defines one profile. The package diagram is used to describe dependencies between profiles (Fig. 3). The class that is included in a diagram from other class diagram is depicted in grey.

WIS configuration profile is defined by the model given in the diagram (Fig. 4). The elements of this model are *Function group, Function*, and *Transition*. The *Function group* consists of many *Functions* implemented in WIS to support business functions of the organization. The function groups have *Transitions* defined between different instances of these groups. The transitions have allowed directions from one function group to another. The directions of transitions are modelled by relationships *From* and *To*.

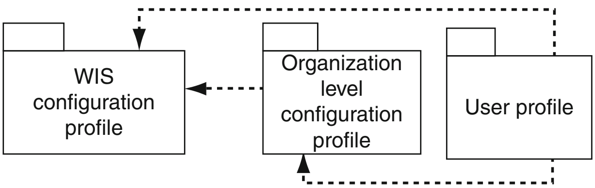


Fig. 3 The package diagram of WIS profile

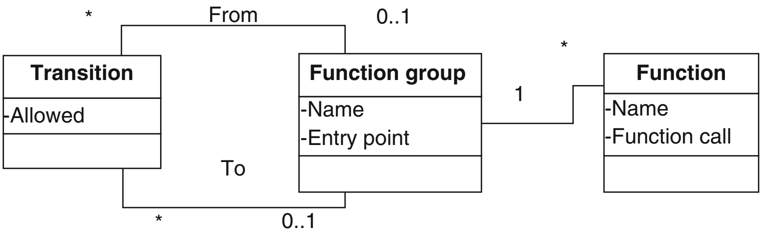


Fig. 4 Configuration profile of WIS

Function group is a grouping of WIS functions, which is characterized by a common *entry point*. The entry point is defined by entry location and entry type. Entry location in this case could be the procedure call in the browser, where the user chooses one of the links representing many entry points into WIS, and which activates a kind of login form. Each login form defines the entry type to the system. For example, there may be different entry types for the authorized part of the system (the database authentication) and for the functions performing self-service tasks (the LDAP authentication).

Depending on the entry point, a joining of the function groups is defined to ensure for the user the option to access many function groups at once without additional navigation, login, and authentication. The joining of function groups is performed based on the allowed transitions defined in the configuration profile of the system.

Let FG1 and FG2 be two function groups. Two transitions between FG1 and FG2 are possible: the transition FG1 à FG2 from FG1 to FG2 and the transition FG2 à FG1 from FG2 to FG1. Which one of them is allowed, it is determined by the *Transition* attribute "allowed". The joining of two function groups could be performed, if the entry point into the function group will be used, from which the allowed transition starts. It could be defined joining FG1 and FG2 in one direction, but not in the opposite direction.

The configuration profile for the organization level instance of the WIS is defined by the model given in the diagram (Fig. 5).

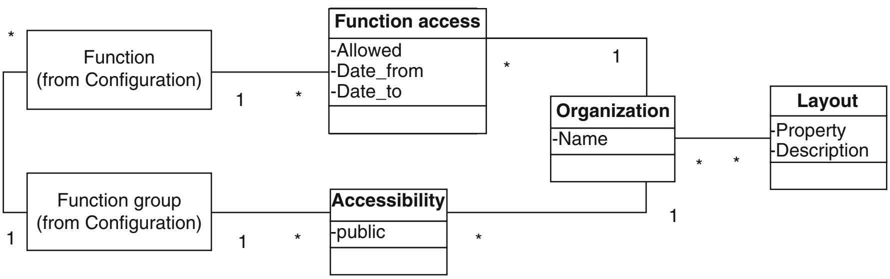


Fig. 5 The configuration profile for the organization level instance of the WIS

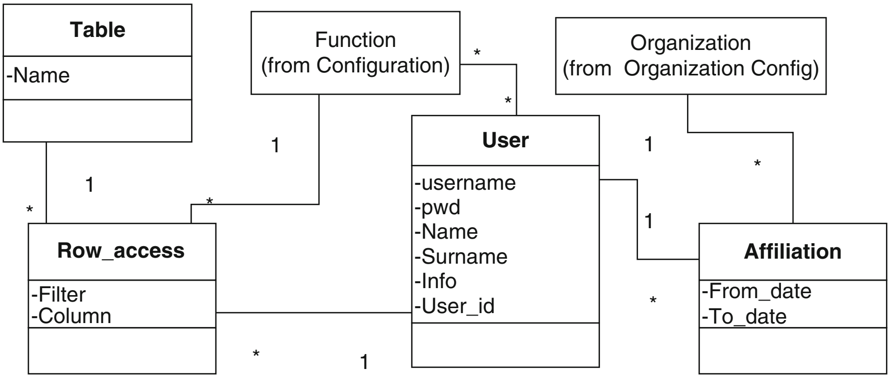


Fig. 6 User profile of WIS

For each particular organization, this profile describes the individual properties (e.g. local configuration, layout etc.) essential for the organization level instance of the WIS. The elements of the model are *Organization, Function access, Accessibility,* and *Layout*. The model uses also classes from configuration profile of WIS (*Function* and *Function group*).

Organization represents those business organizations that use the WIS. *Accessibility* defines for the organization a WIS instance describing, whether access to particular function group is allowed or not to this organization. A function group could be defined as *public*. *Function access* defines for the organization, which functions are accessible in predefined time periods in the case, when the corresponding function group also is allowed. *Layout* describes properties of some elements, if a personalized look of the organization level instance of the WIS is needed. These elements are such as logo, background, font etc.

The user profile is defined by the model given in the diagram (Fig. 6). The user profile describes the affiliation of the user in the particular time period in one or more organizations using WIS. This description defines for the user the corresponding one or more organization level WIS instances. Additionally the user profile describes the user access rights to the functions and data restrictions in the WIS database. The elements of the model are *User, Affiliation, Table*, and *Row_Access*. The model uses also one class from configuration profile of WIS (the class *Function*) and one class from the Organization profile of WIS (the class *Organization*).

User class describes the personal information of the user, login information, and *user_id* that is used to identify the data about the user in the WIS data base. Other data accessible to the user are described by *Row_access* and *Table*. The data restrictions are defined by column filters for the tables. The *Affiliation* describes the affiliation of the user in one or more organizations in the time period. The user has access to the *Functions* within the organization level instance of the WIS.

2.3 Adaptation Component

The adaptation component performs adaptation in two levels: a coarse grained and detailed adaptation. The result of the adaptation is a WIS instance adapted in two levels:

- The result of coarse grained adaptation is an organization level WIS instance adapted to the needs of the particular organization. The adaptation result has also a joined function groups from different WIS instances, if a user belongs to more than one organization using the WIS.
- The result of detailed adaptation is a WIS instance adapted for the individual needs of the user, which is characterized by personalized navigation and personalized content within the framework of the organization level instance of WIS.

2.3.1 Coarse Grained Adaptation

The adaptation component receives from the context monitor the information about the context of the system usage: user, entry point, and time. The adaptation component gets the profiles from the data layer according to the given context: the WIS configuration profile, the configuration profile for the organization level instance of the WIS, and user profile. The coarse grained adaptation is performed by the following operations:

- The construction of initial configuration of WIS
- The WIS adaptation in organization level
- The joining of instances of function groups
- The selection of allowed function groups
- The adaptation of layout

Let us denote the operation "The construction of initial configuration of WIS" with *start_config*. The initial configuration (IC) of WIS is formed according to the configuration profile of WIS. IC consists of all Function groups described in the profile.

Let us denote this operation "The WIS Adaptation in Organization Level" with *adapt_org* (*IC, user, time*). The goal of this operation is the adaptation of IC according to the organization level profile. The operation *adapt_org* indicates the organization from the user profile, which corresponds to the user identified by the context monitor. Then the IC is checked, if all function groups and functions are accessible to the particular organization. The adapted instance WIS_org is the result of this operation.

If the user belongs to more than one organization, then according to the profile of the user, adapted instances are made for each such organization: e.g. *WIS_org1* and *WIS_org2*. For each organization of the user, the *adapt_org* operation is performed separately.

The operation "The Joining of Instances of Function Groups" is necessary in the case, when the user belongs to more than one organization and the corresponding WIS instances, e.g. *WIS_org1* and *WIS_org2* are made. Let us denote the joining operation with *FG_union (WIS_org1, WIS_org2)*. *FGj_1* and *FGj_2* will be the two instances of the arbitrary Function group FGj in two WIS instances *WIS_org1* and *WIS_org2*.

FG_union is defined only for instances of the same function group, and only in the case, when the user belongs to both organizations having *WIS_org1* and *WIS_org2*. The joined instance *WIS_org1_org2* will consist of such joined function groups and Function groups, which are defined in the WIS organization profile only for one organization.

The operation "The Selection of Allowed Function Groups" is necessary in the case, when the user belongs to more than one organization and *adapt_org* and *FG_union* operations are performed. After that a selection of Function groups, according to the entry point is conducted. Let us denote the group selection operation with *FG_select (entry point)*.

Let us assume that an entry point belongs to a function group *FGk*. Let *FGi* be an arbitrary function group; *Tij* be a transition from *FGi* to *FGj*, defined in the configuration profile of WIS. Then it is possible to define a transition chain *TCk*, which consists of sequential transitions *Tij* between function groups starting with function group *FGk*. The number of different possible *TCk* is predefined with the configuration profile of WIS.

The result of *FG_select (entry point)* is the adapted joined WIS instance *WIS_org1_org2*, which contains all function groups that are accessible with any of transition chains *TCk*.

Let us denote with *adapt_layout* the operation "The Adaptation of Layout", which performs the adaptation of the layout of WIS for the needs of the organization. When the user belongs to one organization, the layout properties are found out from the configuration profile for the organization level of WIS and applied to the WIS instance. In the case, when the user belongs to more than one organization, the layout properties are used according to the organization, which has the entry point used by the user.

2.3.2 Detailed Adaptation

The detailed adaptation is based on the user profile. The more detailed adaptation is made for the WIS instance that is already adapted by operations of the coarse grained adaptation. The detailed adaptation has two steps:

- detailed adaptation of the navigation,
- detailed adaptation of the content.

Let us denote these steps with *adapt_navig* (user) and *adapt_data* (user).

The operation *adapt_navig* (user) corresponding to the parameter "user" finds out the user rights to the functions. Within the framework of the adapted WIS instance gained during the coarse adaptation, an adapted navigation according to user rights is created. The adapted navigation provides access to allowed functions using links. We can say that an adapted instance with an adapted navigation of WIS for the user "WIS_user" is made.

The operation *adapt_data* (user) corresponding to the parameter "user" finds out the user restrictions to the data, which are defined using tables, their columns and filters for these columns. The WIS instance with the adapted navigation "WIS_user" is supplemented with the content, which in general case is defined by the data usage of the functions. The operation *adapt_data* (user) adapts the content corresponding to the restrictions defined for the user.

This approach could be used, when simple restrictions should be defined. More complicated adaptation rules are defined and stored in the Base for Adaptation Rules. The adaptation rules could be defined for example with OCL statements. These adaptation rules also define the data restrictions for the users.

3 The Case Study of WIS Adaptation

We will explain the usage of the adaptation architecture in the WIS for Universities (UWIS). Three function groups are defined in the configuration of the UWIS:

- We will denote with *FGa* the group of authorized functions. The authorized functions are the functions that are provided to support the administrative functions of the universities. These functions perform operations with the data of students and employees, e.g. "Entry of student grades", "Change of student statuses", etc. To perform these functions, the users should have access to data of other employees or students within the responsibilities of users' work.
- We will denote with *FGs* the group of self-service functions. These are functions, which the user can perform only with his personal data or data, which are connected with his data. For example, the user can apply for the study courses, he can control the actual status of his grades, statuses, etc.
- We will denote with *FGp* the group of public functions. These are accessible to all users without authentication and usually provide access to different reports that include public data.

The configuration profile of organization level instance of UWIS contains information about logo and other layout elements.

Let us consider a case, when the user is an employee at the Liepaja University (LiepU) and he studies at the University of Latvia (LU). Both universities use UWIS. UWIS has many entry points depending on the number of participants of UWIS project. The context monitor identifies the entry point, the user, and the time, when the UWIS is accessed. The adaptation component finds out that the user belongs to two universities and constructs two instances of UWIS with all three function groups according to UWIS configuration profiles of LU and LiepU (Fig. 7).

The instances of UWIS contain also the predefined transition between function groups, the arrows denote the directions of transitions.

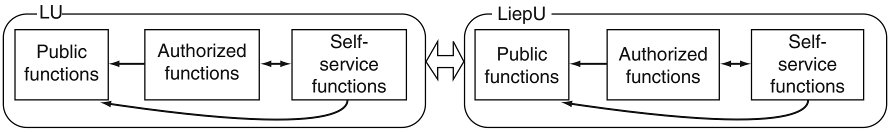


Fig. 7 WIS instances for the case study

4 Results, Conclusion and Further Work

Our architecture is successfully implemented and is being used in two WIS. The proposed architecture of WIS adaptation is used in the project, where 12 universities are using each an adapted instance of the system. All universities (12) use the authorized function group, 8 universities use the public part of the system, but 10 universities use the self-service functions.

The picture (Fig. 8) shows all entry points into the system, including one common entry point into authorized function group and entry points into self-service function groups of each particular university. These entry points into self-service groups serve also as entry points to public functions of each university.

The other implementation is used in the Road Safety Directorate and insurance companies for the car registration WIS.

Our architecture is not an exception among many architectures and approaches discussed previously. We also provide a little different view on the problem of WIS adaptation, looking from perspective of organizations that are interested in an adapted WIS for their needs, if a unified system to support similar business processes is used. We introduce two levels of adaptation—coarse grained adaptation for the organization and fine grained adaptation for the user, we also support the situation, when users can work with many instances of the system adapted for different organizations, which are integrated into one instance for a particular user.

Some aspects of adaptation provided in the architecture were out of the scope of this paper, e.g. the process execution monitoring to provide more effective WIS and to

Fig. 8 Entry points into function groups in the UWIS (LAIS in Latvian)

find out new possibilities for the adaptation. We plan to use our work in process measurement and supplement it with adaptation aspects. Also more complicated adaptation scenarios that use adaptation rules should be formalized and described in detail.

Acknowledgements This work has been supported by ESF project No. 2009/0216/1DP/1.1.1.2.0/09/APIA/VIAA/044.

References

1. Fraternali P (1999) Tools and approaches for developing data intensive web applications: a survey. J ACM Comput Surv 31(3):227–263
2. De Virgilio R, Torlone R (2005) A general methodology for context-aware data access. In: Proceedings of the 4th ACM international workshop on data engineering for wireless and mobile access, MobiDE'05, ACM, New York, pp 9–15
3. Abbar S, Bouzeghoub M, Kostadinov D, Lopes S, Aghasaryan A, Betge-Brezetz S (2008) A personalized access model: concepts and services for content delivery platforms. In: Kotsis G, Taniar D, Pardede E, Khalil I (eds) Proceedings of the 10th international conference on information integration and web-based applications and services, iiWAS'08, ACM, New York, pp 41–47
4. Valeriano D, De Virgilio R, Torlone R, Di Federico D (2006) An efficient implementation of a rule-based adaptive web information system. In: Frasincar F, Houben G-J, Thiran P (eds) Proceedings of international CAISE workshop on web information systems modeling (WISM'06), CEUR Workshop Proceedings, vol 239
5. De Bra P, Houben GJ, Wu H (1999) AHAM: a dexter-based reference model for adaptive hypermedia. In: Proceedings of the 10th ACM conference on hypertext and hypermedia hypertext'99, ACM Press, New York, pp 147–156
6. Tvarožek M, Barla M, Bieliková M (2007) Personalized presentation in web-based information systems. In: Leeuwen J, Italiano GF, Hoek W, Meinel C, Sack H, Plášil F (eds) Proceedings of the 33rd conference on current trends in theory and practice of computer science. LNCS, vol 4362, Springer, New York, pp 796–807
7. Jablonski S, Petrov I, Meiler C, Mayer U (2004) Guide to web application and platform architectures (Springer Professional Computing). Springer, New York
8. Clements P, Northrop L (2002) Software product lines: practices and patterns. Addison-Wesley, New York
9. Bass L, Clements P, Kazman R (2003) Software architecture in practice, 2nd edn. Addison-Wesley, New York, p 560
10. Microsoft (2006) Architecture strategies for catching the long tail, April 2006

Change Work in Organisations

Some Lessons Learned from Information Systems Development (ISD)

Anders G. Nilsson

Abstract In change work we have the ambition to improve or enhance different activities within a specific situation or context. We can think of e.g. changes in society, in organisations or in family life. In this case the focus will be on change work in organisations; private companies as well as in public services. Change work implies a purposeful growth and development of organisations. This development work can be performed by operating in networks (inter-organisational change) or accomplished by undertaking individual measures (intra-organisational change). By information systems development (ISD) we mean analysis, design and implementation of useful IT systems in companies. ISD is nowadays regarded as an essential and vital part of change work in organisations. This paper presents ten lessons learned from working with information systems development in a change work context.

1 Information Systems Development (ISD)

By information systems development we mean analysis, design and implementation of useful IT artifacts to support some kind of business in organisations (Orlikowski and Iacono 2001). By IT artifacts we mean the use of hardware and software solutions to improve the business activities within and between organisations. The IT artifacts can be of a varied character—for example we can create information systems in organisations by using bespoke (tailor-made) software, application packages or component-based solutions. We are here focusing on computer-based systems for developing and changing the situation in concrete business cases.

Research on information systems development (ISD) has its roots back in the mid 1960s. Scandinavian researchers have had a great influence on the evolution of information systems as an academic discipline (see Iivari and Lyytinen 1998). Personally, I had the privilege of being a member of the Scandinavian school and tradition of information systems development (Langefors 1973, 1995). My main

A. G. Nilsson (✉)
Karlstad University, SE-651 88, Karlstad, Sweden
e-mail: anders.nilsson@kau.se

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_44, © Springer Science+Business Media, LLC 2011

experiences are based on working with the ISAC approach for requirements specifications (Lundeberg et al. 1981), the SIV method for purchasing standard application packages or ERP systems (Nilsson 2001) and the Business Modelling framework for studying method combinations (Nilsson et al. 1999). After practicing in the ISD area for nearly 40 years, as a researcher/teacher (for academia) and advisor/counsellor (to industry), I feel a great need to offer some reflections on my findings.

My perception of the ISD field can be described in many different ways. In this case, I have chosen to present my strongest impressions from working with information systems development as lessons learned for change work in organisations. This is an attempt to explore essential patterns or fundamental principles for business development grounded in theory and practice. The lessons learned are summarised in 10 points and the order between them is approximately how they appeared to me over the years.

2 Lesson 1: Proceed from User Needs

Change work in organisations can be seen as a social field of forces between different interest groups or stakeholders such as general managers, business people and systems designers. There exist from time to time communication gaps or misunderstandings when people from these interest groups try to deal with development matters (Nilsson et al. 1999). Therefore it is important to find ways to bridge the communication gaps between key actors during change work or business development.

From the ISD area we have learnt the lesson to proceed from the user needs, requirements and terms during the development work. The simple argument is that there are the real users (or business people) who in their daily work should live with the proposed changes, e.g. new information systems. The principle of user orientation goes back to professor Börje Langefors' infological approach to information systems development. The theory of infology states the significance of designing and operating information systems from a user point of view in order to achieve desired results in organisations (Langefors 1973, 1995). The ISAC approach for systems development was strongly built on a user perspective for change work (Lundeberg et al. 1981). The principle behind socio-technical design emphasises the importance of integrating people and technical matters during systems work (Mumford 1971). An interesting observation is that the same reasoning lies behind the well-known success formula created by Likert (1961, p. 212):

$$\text{Degree of success in change work} = \text{f}(\text{Quality} \times \text{Acceptance})$$

The success formula states that to attain a successful result, we must have both sufficient quality in the designed solutions (e.g. the IT artifacts), and a good acceptance among the co-workers (users) to give them a motivation for using the solution. A low value in either quality or acceptance will lead to an unsuccessful result—hence the multiplication sign in the formula!

3 Lesson 2: Apply Methods and Models

Change work in organisations usually implies a comprehensive and complex task in dealing with the above mentioned stakeholders or interest groups. We need to make many different decisions on a huge number of issues during development processes. We need to take care of a variety of users and their mental models (images) of business performance. Business and systems development often involve people from different application areas (such as production, marketing, accounting) and they have various perceptions of the present situation—they may stress several desires and requirements that can be overlapping or contradictory in character. We can also have communication problems between business people (users) and systems people (designers) during the development work. How can we professionally handle such problems?

One lesson learned from the ISD field is that development work is performed more efficiently with support from formal methods and models. In this sense a specific method (or approach) can be a useful tool in creating a common language between general managers, business people and designers. Another basic principle in the infological approach to systems development postulates that applying methods and models is a very sharp way of representing users' needs and requirements (Langefors 1995). The ISAC approach was one of the first methods in the world that tried to show a systematic method from problem capturing (so-called change analysis) down to creation of technical solutions (data/program design). See Lundeberg et al. (1981).

By method, we mean concrete guidelines or prescriptions for a systematic way of working with development tasks in organisations. It is possible to distinguish between three main constituents of a method (Nilsson 1995): Perspectives (basic principles and assumptions), Work Model (steps and documentation) and Interest Group Model (stakeholders and collaboration forms). There has been much debate over the years about the actual effects of method use in practice. Below we summarise some essential needs for applying methods and models to support development work in organisations:

- *Requirements specifications;* the ultimate need to produce an exact, consistent and complete requirements specification for designing the future operations.
- *Explain IT possibilities;* the need for explaining how new IT possibilities (e.g. e-business, Internet) can enhance business operations and corporate strategies.
- *Describing business flow;* the need for describing and coordinating the complex nature of material flow, service flow, information flow and cash flow.

Applying methods and models for a systematic way of working with development issues should be adapted to the special situation for change in organisations. There is a trend today to use tool-boxes or tool-kits consisting of a various selection of methods for different situations (Nilsson 1999). We have not yet found a "super methodology" in order to attack or handle all possible development tasks during change work. Therefore it is important to try to combine separate methods from different fields like accounting (Samuelson 1980), service marketing (Edvardsson et al. 2000), strategic management (Kaplan and Norton 1996) together with our ISD methods (Avison and Fitzgerald 2003).

4 Lesson 3: Consider Different Perspectives

Change work in organisations will give better results if we turn the descriptions or models of the business operations over in our minds. By changing perspective and observing our operations from many different angles, we will gain a deeper understanding of the underlying mechanisms in the organisation. Thus by doing so, we will have a more solid base from which to suggest vigorous changes in the business operations. It is therefore important to consider different perspectives when we try to understand and change the business situation in our organisations. Lundeberg (1993) proposes a combined perspective approach to manage change processes in business.

Another lesson learned within the ISD area is that methods for systems work should emphasise different perspectives or aspects when describing business operations and their supporting information systems (cf. Olle et al. 1991; Sowa and Zachman 1992). Requirements specification is an instrument for accurate descriptions of the contributions and effects a specific IT artifact needs to provide to the business operations. The specification should illustrate different users' demands on the new business and systems solutions. One problem with requirements specifications has been that they are one-dimensional in character, illuminating only a very limited perspective of business and information modelling. In development work we need to describe business and information operations from many different perspectives, such as (Nilsson 1995, 2001): *Intentions* (concerning goals, visions, problems and strengths, etc.), *Activities* (concerning functions, processes, and tasks, etc.), *Resources* (concerning data, concepts, components and objects, etc.), and *Behaviours* (concerning events, rules, actors, and force fields, etc.).

In the early days of the ISD field, different method schools were in competition with each other. Each of them represented only one of the four above mentioned perspectives. Today we more and more seek suitable combinations for development work. In real life projects these four basic aspects need to complement each other. Hence we need an appropriate mixture of perspectives depending on the situation at hand (cf. Yourdon 1993; Booch et al. 1999).

5 Lesson 4: Understand Current State, Change to Better Situation

Change work in organisations is often performed in parallel with the day-to-day operations in a going concern. An understanding of the current state of business gives a more stable base to create realistic changes for achieving a better situation in the future. Lundeberg (1993) proposes to use time frames when working with

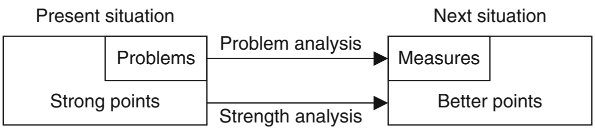


Fig. 1 From present situation to next situation

change processes in business. He distinguishes between observing the organisation from a past situation, to a present situation and into a future situation. In practice a historical review of earlier situations or milestones when developing organisations is seldom done. At best we take care of the past experiences when analysing the present situation. A classical problem in change work in organisations is illustrated by Fig. 1 and explained below:

We are often focused, in change work, on practical problem solving or "troubleshooting" when it can be better to look for strong points in our business situation. We are educated to make a professional analysis of the problem complex and then to find clever business solutions or smart development measures. But in many cases the problematic situation may represent only 25% of our current state of business. The mistake is that we don't focus on strengths and opportunities to make an even better situation for future operations. The strong points should be taken care of and sharpened and improved before entering the next situation.

A lesson learned from the ISD field is the importance of starting the development work with a careful pre-study to identify problems and strengths in the present situation as a platform for suggesting appropriate development measures. The ISAC approach became an image or ideal type for other ISD methods as regards separating the development work into two main phases: first a change analysis stage before starting up an information systems project (Lundeberg et al. 1981). The business process management movement also focuses on different states or situations (cf. Davenport 1993). These approaches separate between making process models for the "is-state" and the "should-state" for our business. By making "is-models" you are able to understand the organisation in its current situation before making "should-models", in order to be competent to change to a better situation (Tolis and Nilsson 1996).

6 Lesson 5: Use Appropriate Enablers

Change work in organisations can be driven by using different opportunities, or limited by various constraints. Human and technology factors can be both opportunities and constraints for change, depending on the situation. Davenport (1993) uses the term "enablers" for possible drive forces for change. The human factors represent the capabilities offered by knowledge, skill and motivation of the co-workers in the whole organisation. The technology factors represent the capabilities offered by human information, computer solutions, software applications, telecommunications, etc. Personal knowledge development and technological innovations can be good enablers when changing the business situation in organisations. Ploom (1988) describes a change model where an organisation goes through three consecutive phases: (1) the efficiency phase, (2) the integration phase and (3) the transformation phase. The phase of transformation represents the most challenging process where we use different enabling factors to obtain a strategic position for the company in the value chain on the market.

One lesson learned from the ISD area is that systems development efforts have changed focus over the years from designing information systems in order to support business operations (resource approach), to a position where we design information in our computer-based systems to create new business opportunities for the organisation, and hence strengthen the competitive edge on the market (enabling approach). In the first approach the information systems are regarded as resources in change work. Starting with the needs of the users, a business operations specification is made which provides both content and structure requirements on the information systems. In the second approach the information systems are regarded as enablers for change. Here the focus is on the potential that a new information system represents for the organisation. The information system becomes an enabler for renewing the business. New technological innovations in multimedia (cf. Packer and Jordan 2001), the Internet and electronic commerce (Earl and Khan 2001) have become new value-adding enablers to the business.

7 Lesson 6: Time Is Critical for Change

Change work in organisations is often restricted by time in one sense or another. In other words, time is a critical factor for handling change processes. As mentioned above (under Lesson 4), Lundeberg (1993) Proposes to use time frames when working with change processes in business, i.e. distinguishing between past situations, present situations and future situations. During change work in practice we have also to balance between quality and temporal issues in order to gain successful results in time and budget. In change projects we have to deliver acceptable results at agreed deadlines. An interesting observation is that time can be perceived somewhat differently by various stakeholders. For example, there is a phenomenon in practice that can be framed as “threshold levels”. This means that a user demands some specific messages from a reporting system or a data retrieval system at a certain point in time—in this case he/she wants this information neither earlier nor later in time.

In the ISD field we have learnt the lesson that time is very important when developing information systems. It is not an understatement to say that Langefors (1973) in his theoretical and empirical work has “reinvented” the significance of the time concept for successful systems design. In the infological approach to information systems development we can find three circumstances where time has to be considered in a clear and explicit way (Langefors 1995). Firstly, we have to strive for executive optimal solutions or sufficiently good information systems with regard to the user needs, together with time and cost limits for implementation. It is in this sense a trade-off between information needs and time restrictions. Secondly, we have to consider the infological equation where the time component is essential for a user to be able to interpret personal information from a given set of data. The infological equation states: I = i (D, S, t) where "I" is the information conveyed, "i" is the interpretation process, "D" is the data at hand, "S" is the pre-knowledge, frame reference or mental structure of the user, and "t" the time required or available for the process. When a user needs more time for interpretation it could mean losses in efficiency. Thirdly, we have to consider how messages should be designed for a better understanding and communication. An elementary message (e-message) represents the smallest information unit in a system and is defined as the following triplet: object, time, property. It says that such an elementary message describes a property (e.g. price) for a specific object (e.g. article) at certain time (e.g. year-month-day). The reasoning behind this principle for systems design is that it is urgent with time stamps for messages in order to avoid confusion in operating future information systems.

8 Lesson 7: Pay Attention to a Life Cycle Philosophy

Change work in organisations goes through a life cycle with sequential, parallel and/or iterative phases. It is the same way with change processes as with e.g. product and market development processes. A life cycle can be partitioned in a number of phases or areas. According to Nilsson (1995), on a crude level a development process can consist of phases for change analysis (with enterprise models), formulation (of requirements specification), implementation (of business solution) and after some time assessment (review of business operations). These phases or areas focus on different kinds of problems and demand various bodies of knowledge and competence. What pattern lies behind a life cycle philosophy? Development work can be seen as a form of decision-making activity. Simon (1965) states that all kinds of decision-making go through three phases: intelligence (I), design (D) and choice (C). When we come to the situation to carry out or execute a decision it is according to Simon again a decision-making activity (with its own IDC triplet). Lundeberg (1993) describes a general model for change processes (based on IDC) comprised of three recurrent and overlapping phases: planning (goals), operation (activities) and evaluation (evidence).

What we have learnt as a lesson from the ISD area is that it is fruitful to consider a system's life cycle consisting of phases for acquisition, use, maintenance and phasing-out. Strictly speaking, by information systems development we mean the acquisition phase including steps for analysis, design and implementation of IT artifacts (cf. Hawryszkiewycz 2001). The life cycle for creating and managing information systems in organisations has over the years shown to be an essential and valid concept, and therefore it forms an important basis for construction of methods for systems work. In 1967, professor Langefors worked out and presented an original proposal for partitioning of the system's life cycle. The result was four classical problem areas which have had a great impact on subsequent development of ISD methods and approaches (Langefors 1974): (1) object system analysis and design, (2) information analysis, (3) data system architecture and construction, and (4) realisation, implementation and operation. The two first areas treat infological or user-oriented problems, while the two last areas treat datalogical or technical problems. The traditional ISAC approach was built on these four classical problem areas within information systems development (Lundeberg et al. 1981).

9 Lesson 8: Reuse Successful Solutions

Change work in organisations should be more effective if we can gain access to past experiences formalised in e.g. best practice models, application templates and/or standardised solutions. These represent generalised experiences of a certain business or application domain. As a concrete example we can mention the RP model as a framework for designing accounting information systems (Samuelson 1980). Experience of good ideas and strong points from successful business cases should be taken care of when designing for future situations. A possible alternative is to reuse knowledge by purchasing requirements specifications from outside instead of acquiring ready-made or pre-specified solutions. In this case, we build our competence for carrying out development work on a higher level of abstraction—established and proven knowledge—rather than on fixed or "quick and dirty" attitudes. During change work we can use or create solutions with different degrees of "pre-specification". By this we mean how complete specifications we have in advance when the development work starts (Nilsson 1995). A development situation which gives us an opportunity to gain a higher degree of pre-specification facilitates the possibilities for reuse of successful solutions!

From the ISD field we have learnt a lesson to use standard application systems (Nilsson 2001) and ERP-systems or enterprise systems (Davenport 2000) as an efficient way of reusing successful solutions. The degree of success for implementation of such kind of ready-made software as IT artifacts in organisations depends on how well-prepared the managers and users are for this new business challenge. A careful vendor assessment is an important work task during the acquisition process. Obtaining new package releases from the vendor are critical issues for the work with maintenance management. Another essential trend in the ISD field is the phenomenon labelled object-oriented or component-based systems development (Jacobson et al. 1993). Business objects or components here represent application parts in miniature. This way of working with objects/components is an approach for reuse in a small scale compared to the situation with standard application systems which is reusing solutions in a larger scale. Today, traditional ERP vendors try to renovate and reconstruct their old packaged software using object or component development techniques as a competitive weapon.

10 Lesson 9: Discover Reality by Prototyping

Change work in organisations should in practice be carried out with the help of some systematic model of planned activities (see Nilsson 1995). The initial change models for development work, presented in the market, were sequential in nature. They were often labelled "waterfall models" meaning that a certain phase must be finished before the next phase can start. But change work is seldom strictly sequential or linear in character. Therefore new change models were presented in the market proposing that development work should be performed in a number of partly overlapping phases. They were labelled "sliced models" meaning that certain phases can be done in parallel. But change work in practice is not so often purely sequential and/or parallel in character. Therefore new change models were presented again in the market proposing that development work can be iterative in nature. They were labelled "prototyping models", meaning that certain phases should be recurrent as new knowledge appears during later work. By making rapid prototypes of a desired future business situation in reality or every-day life, the various interest groups have the possibility to discover the effects of introducing new types of solutions (e.g. IT artifacts). The stakeholders can react to the prototype solutions and give valuable feedback for further specification of the new business situation. A prototype gives a concrete picture of a business solution and implies a rich learning environment for the managers and users who participate in and run the change processes.

Already at an early stage in the evolution of the ISD area we learned a lesson that users need to experience a prototype or realistic systems sketch before they could describe the exact requirements on a new information system (Bally et al. 1977). Prototypes can be designed in various forms, all the way from simple "tear and wear" solutions to more advanced pilot systems expandable to "full-scale" solutions (cf. Budde et al. 1992). The argument for prototyping in ISD work is that an IT artifact is perceived as a rather complex phenomenon. We can not therefore always plan for new information systems in a strictly analytical way (in the model world) but we need also to do some practical experiments (in the real world) gradually. In analytical systems development the requirements specification from the users needs to be complete and "frozen" before an implementation of the information system may begin. In experimental systems development (prototyping) there is interplay between the work with specification and implementation, sometimes in several rounds. The prototyping approach has not always produced the desired effects in organisations, since we seldom are capable of supplementing the experimental work with a solid evaluation phase. Close to the ISD field is the multimedia area. When developing multimedia systems, the prototyping approach gives the stakeholders a deeper understanding of how the interactive media product would work in practice (cf. Packer and Jordan 2001).

11 Lesson 10: Promote Business in Manageable Steps

Change work in organisations means that we are advancing the business towards some concrete visions or goals. There are many different types of change programs in practice. Business process reengineering (BPR) implies work with radical changes to achieve dramatic improvements in business performance (see e.g. Davenport 1993). In other words, we strive for quantum leap process improvements with immediate results. Total quality management (TQM) implies work with incremental changes to gradually achieve better results in business performance (see e.g. Ishikawa 1985). In other words, we strive for continuous process improvements from time to time. We can also think of mixed forms between these two extreme types of change programs. One possible example of such a change program can be called business process elevation (BPE). Here we try to make business improvements in distinct and manageable steps (see e.g. Nilsson et al. 1999). The size of changes required in business operations depends on the specific situation. In other words, we strive for promoting the business on a regular basis in order to achieve our visions and goals. Below we launch a change model for business promotion in line with a BPE philosophy. See Fig. 2 below.

A lesson learned from the ISD field is the significance of starting up development work from a change analysis which builds a platform for further development of e.g. information systems. The model for promoting business operations is based on the change analysis method in the traditional ISAC approach (Lundeberg et al. 1981). The change model can be regarded as a clock starting with a goal analysis for the organisation (12 o'clock). We then move on with strength analysis, problem analysis and stakeholder analysis, i.e. people (users) who are affected by the problems and strengths. These analyses build a platform for assessing the present situation before making a "brain-storming" session with an improvement analysis where we generate appropriate change measures. Again we make an assessment but now for the next situation for the organisation. Thereafter it is time for the implementation phase when we introduce the desired business changes in daily work. After a period of time we start a new change program for business promotion according to the clock model.

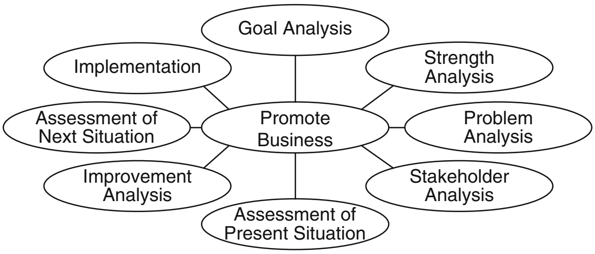


Fig. 2 A change model for promoting business—the clock model

References

Avison DE, Fitzgerald G (2003) Information systems development: methodologies, techniques and tools, 3rd edn. McGraw-Hill, London

Bally L, Brittan J, Wagner KH (1977) A prototype approach to information system design and development. Inf Manag 1:21–26

Booch G, Rumbaugh J, Jacobson I (1999) The unified modelling language user guide. Addison-Wesley, Reading, MA

Budde R, Kautz K, Kuhlenkamp K, Züllighoven H (1992) What is Prototyping?. Inf Technol People 6(2–3):89–95

Davenport TH (1993) Process innovation: reengineering work through information technology. Harvard Business School Press, Boston, MA

Davenport TH (2000) Mission critical: realizing the promise of enterprise systems. Harvard Business School Press, Boston, MA

Earl M, Khan B (2001) E-commerce is changing the face of IT. MIT Sloan Management Review, Fall 2001, 64–72

Edvardsson B, Gustafsson A, Johnson MD, Sandén B (2000) New service development and innovation in the new economy. Studentlitteratur, Lund, Sweden

Hawryszkiewycz I (2001) Systems analysis and design, 5th edn. Prentice Hall/Pearson Education Australia, Sydney

Iivari J, Lyytinen K (1998) Research on information systems development in scandinavia – unity in plurality. Scandinavian Journal of Information Systems 10(1–2):135–186

Ishikawa K (1985) What is total quality control? The Japanese Way, Prentice-Hall, Englewood Cliffs, NJ

Jacobson I, Christerson M, Jonsson P, Övergaard G (1993) Object-oriented software engineering: a use case driven approach, 4th revised printing. Addison-Wesley, Wokingham, England

Kaplan RS, Norton DP (1996) The balanced scorecard: translating strategy into action. Harvard Business School Press, Boston, MA

Langefors B (1973) Theoretical analysis of information systems {THAIS}. Studentlitteratur, Lund and Auerbach, Philadelphia, PA

Langefors B (1974) Information systems. IFIP Congress 74. Stockholm, North-Holland Publishing Company, Amsterdam, pp 937–945

Langefors B (1995) Essays on infology: summing up and planning for the future. Studentlitteratur, Lund, Sweden

Likert R (1961) New patterns of management, McGraw-Hill, New York

Lundeberg M (1993) Handling change processes: a systems approach. Studentlitteratur, Lund, Sweden

Lundeberg M, Goldkuhl G, Nilsson AG (1981) Information systems development: a systematic approach. Prentice-Hall, Englewood Cliffs, NJ

Mumford E (1971) Systems design for people: economic evaluation of computer based systems. The National Computer Center (NCC), Manchester, England

Nilsson AG (1995) Evolution of methodologies for information systems work: a historical perspective. In Dahlbom B (ed) The infological equation: essays in honor of Börje langefors, Gothenburg studies in information systems. Göteborg University, Göteborg, Sweden, pp 251–285

Nilsson AG (1999) The Business Developer's Toolbox – Chains and Alliances between Established Methods. In Nilsson AG, Tolis C, Nellborn C (eds) Perspectives on business modelling: understanding and changing organisations. Springer, Berlin, pp 217–241

Nilsson AG (2001) Using standard application packages in organisations: critical success factors. In Nilsson AG, Pettersson JS (eds) On methods for systems development in professional organisations: The Karlstad University approach to information systems and its role in society. Studentlitteratur, Lund, Sweden, pp 208–230

Nilsson AG, Tolis C, Nellborn C (eds) (1999) Perspectives on business modelling: understanding and changing organisations. Springer, Berlin

Olle TW, Hagelstein J, Macdonald IG, Rolland C, Sol HG, Van Assche FJM, Verrijn-Stuart AA (1991) Information systems methodologies: a framework for understanding, 2nd edn. Addison-Wesley, Wokingham, England

Orlikowski WJ, Iacono CS (2001) Research commentary: desperately seeking the 'IT' in IT research – a call to theorizing the IT artifact. Inf Syst Res 10(2):121–134

Packer R, Jordan K (eds) (2001) Multimedia: from wagner to virtual reality. W.W. Norton & Company, New York

Ploom A (1988) Information technology and the manufacturing enterprise. NordDATA 88, Helsinki, Band 2, pp 275–281

Samuelson LA (1980) Models on accounting information systems: the Swedish case. Studentlitteratur, Lund, Sweden

Simon HA (1965) The shape of automation for men and management. Harper and Row, New York

Sowa JF, Zachman JA (1992) Extending and formalizing the framework for information systems architecture. IBM Syst J 31(3):590–616

Tolis C, Nilsson AG (1996) Using business models in process orientation. In Lundeberg M, Sundgren B (eds) Advancing your business: people and information systems in concert, Chapter VIII. Stockholm School of Economics and Studentlitteratur, Lund, Sweden

Yourdon E (1993) Yourdon systems method: model-driven systems development. Yourdon Press, Prentice-Hall, Englewood Cliffs, NJ

Towards Understanding Contradictions in Enterprise System Implementations—Insights from a Case Study

Stig Nordheim

Abstract This paper presents findings from a study of the implementation of an enterprise system in an organization. The implementation process is viewed from a dialectic perspective, which means thinking in terms of contradictions. This paper raises the following research question: How can we understand contradictions in enterprise system (ES) implementations? To answer this question, an interpretive research approach was chosen. The empirical part is a longitudinal case study. The system was in this case an innovative combination of collaboration and information management technologies. The main contradiction studied in this case was between an as-is implementation of standard software, and an implementation fulfilling the organizational requirements of solution integration and user experience. To understand the issues involved in this dialectic of adaptation, three different perspectives are applied. These are (i) considering ES vendor challenges, (ii) exploring contradictions in the chartering phase, and (iii) understanding contradictions in the project phase. This paper contributes to understanding how the dialectic of adaptation may emerge, and presents three perspectives for understanding contradictions that may occur as an enterprise system is implemented. This understanding may help to constructively deal with dialectics in future enterprise system implementations. Implications for both research and practice are outlined.

1 Introduction

Implementing an Enterprise System (ES) in an organization is both challenging and expensive (Seddon 2005). In ES implementations value conflicts occur between stakeholders (Allen 2005), and a dialectic perspective thus explains important aspects of the ES implementation process (Besson and Rowe 2001; Nordheim and Päivärinta 2006; Robey et al. 2002; Sia and Soh 2007). A dialectic perspective views change as

S. Nordheim (✉)
University of Agder, Postboks 422, 4604, Kristiansand, Norway
e-mail: stig.nordheim@uia.no

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_45,

the result of contradictory values competing for domination (Van De Ven and Poole 1995).

This study focuses on an ES different from ERP systems. The case is a combination of technical solutions, including an integration of Enterprise Content Management with collaboration solutions and personal productivity tools. A number of software systems were integrated, and dialectics occurred in the implementation process. Having established main contradictions in an ES implementation process, one needs to explore them further in order to understand them. Thus a rather fundamental research question is raised:

> How can we understand contradictions in enterprise system implementations?

To understand and explain contradictions is a necessary prerequisite for being able to deal with them in a constructive way.

This paper is organized as follows: Sect. 2 presents background literature. After a description of the research method in Sect. 3, Sect. 4 describes the case. Section 5 presents the findings on how to understand contradictions. Section 6 discusses the findings, concluding with implications for ES research and practice.

2 Background Literature

The term enterprise system (ES) has traditionally been related to ERP systems (Davenport 1998). In addition to ERP, the range of available enterprise systems is growing and includes customer relationship management (CRM), supply chain management (SCM) and enterprise portals (Ward et al. 2005). Another enterprise-wide system is enterprise content management (ECM) (Päivärinta and Munkvold 2005; Smith and McKeen 2003).

Enterprise systems may in a wide sense be viewed as a response to a need aptly summarized by Markus (1997), who stated that the world around us is demanding that we find ways to satisfy unique needs with generic software and components. This is no small challenge for organizations, as research on ES implementation has established (e.g. Soh and Sia 2005).

2.1 Dialectics

Dialectics is one way to view organizational change, as the result of contradictory values competing for domination (Van de Ven and Poole 1995). Dialectics is a way of thinking that is based on contradictions. A contradiction can be viewed as a relation between two opposite aspects of a phenomenon, called thesis and antithesis; where antithesis is the negation of the thesis. The two aspects of a contradiction are intrinsically related, yet opposite and distinct from one another (Van de Ven and Poole 1995). In dialectical theory, stability and change are explained by reference to the balance of power between the two opposing entities. A thesis (A) may be challenged by an antithesis (Not-A), and the resolution of the conflict becomes a synthesis (which is Not Not-A). By its very nature, the synthesis is a novel construction that departs from both the thesis and the antithesis. This synthesis, in turn, becomes a new thesis as the dialectical process continues. Dialectics is about dynamics, and is one way of explaining development and change (Van de Ven and Poole 1995).

Dialectical reflection is a way to understand a situation (Israel 1979). Dialectical theory provides insights into IS development, but does not treat relationships between organizations and IS as determinate, causal connections. Instead, it examines them as emerging through social constructions (Sabherwal and Newman 2003). The mutual adaptation between the technology and the organization is far from a planned change, and the outcomes of the process are difficult to predict (Wei et al. 2005).

2.2 Commonality and Variability: A Vendor Challenge

One way to achieve the challenge to satisfy unique needs with generic software and components (c.f. Markus 1997) is by means of two fundamental design concepts which are a characteristic of most software packages. That is the distinction between commonality and variability (Bühne et al. 2005). These two fundamental design concepts are also used by ES vendors (Leishman 1999). In a previous study of ES vendors, one major ES vendor challenge was found and this was termed a dialectic of design (Nordheim 2007). This challenge is to find an optimal balance between stability and change, between commonality and variability. For an ES customer, there are two important implications of the challenging dialectic of ES design. First, if ES variability is unable to meet the customer's requirements, a considerable pressure is applied to make the customer adapt to the system. ES vendors were also found to simultaneously promote both commonality and variability in a dialectic rhetoric vis-à-vis a potential customer (Nordheim 2007).

2.3 Understanding Contradictions: Previous Research

One contradiction related to ERP implementation is termed a dialectic of learning (Robey et al. 2002). The dialectic of learning occurs between the old knowledge embedded in business processes and practices associated with legacy systems and the new business processes and practices that ERP is designed to support.

Other research on ERP contradictions is Besson and Rowe's (2001), who found that during and after the shakedown phase, targeted organizational outcomes are often not realized because of job and governance conflicts among the users and other stakeholders. Hence, the dialectical view often dominates the shakedown phase, with task, resource and power conflicts among the stakeholders (Besson and Rowe 2001).

An ES entails many stakeholders who typically have multiple and often conflicting objectives and priorities, and rarely agree on a set of common aims (Sedera et al.

2004). Stakeholders with divergent interests can play a vital role in ES implementation success (Boonstra 2006). Therefore value conflicts occur between stakeholders in ES implementation processes (Allen 2005).

Another stream of research has focused on ERP misfits by applying a dialectic conceptualization (Soh et al. 2003; Soh and Sia 2005; Sia and Soh 2007). According to this research stream a misfit emerges between the features of an ERP package and the specific requirements of an organization adapting the package. This misfit between the ERP's structures and the structures of the implementing organization may be solved either by modifying the package or changing the organization (Soh and Sia 2005).

To summarize the ERP literature, it does provide some answers to the research question. We may understand contradictions in ES implementation as: (i) old knowledge versus new knowledge (Robey et al. 2002), (ii) conflicts between stakeholders in the shakedown phase (Besson and Rowe 2001), (iii) contradictions may be understood as conflicting objectives and priorities between stakeholders (Allen 2005), and (iv) misfits between structures in the ES and in the organization (Soh and Sia 2003; Sia and Soh 2007).

3 Case Background and Description

Statoil is a technology-based international energy company that primarily focuses on upstream oil and gas operations. Statoil's headquarters are in Norway, and the largest shareholder is the Norwegian state with a majority of the shares. The company is the third largest exporter of crude oil in the world, with approximately 25,600 employees in 33 countries at the time of study.

Statoil was one of the world's largest users of Lotus Notes/Domino in the 1990s (Munkvold and Tvedte 2003). They launched a major ECM and collaboration development program from 2002 to 2007, which is the object of study. The project included collaboration and information sharing across organizational and geographical boundaries, with access to corporate information. Each employee in Statoil is attached to team sites to conduct their knowledge work and collaboration. All relevant information for a project or team is accessible to the team sites, with all documents being managed throughout their life cycles.

The technologies that constitute the enterprise system include Meridio ECM combined with Stratify content classification (taxonomy) software. This is closely integrated with the FAST Search and Transfer software. Microsoft SharePoint portal server combined with Microsoft Live Meeting constitute important collaboration software, and personal productivity tools include Microsoft Office. The solution spans a wide range of technologies that are tightly integrated. Statoil employees and partners collaborate in the team sites, and their information objects are managed in a way that is transparent to the user. The team sites, corporate-wide integrated storage and search engines are accessed through a corporate portal. The case is therefore a complex ES solution. For Statoil, this ES implementation project is an effort comparable to their ERP implementation in the 1990s.

The project was organized with the CIO as sponsor, a steering committee, a project group with a project manager, and a reference group with representatives from the business units. In addition, there was a quality assurance group, and a solution handover group. The steering committee consisted of process owners and IS/IT managers from different business units having a customer role. The sponsor represented corporate management, was financially and commercially responsible, and chaired the steering committee. The project group was staffed by corporate services IS/IT. Vendor and other consultants were involved as needed, but Statoil staff did a major part of the work. Informants came from the project group, steering committee, corporate user representatives, reference group and quality assurance group.

3.1 Data Collection and Analysis

Some of the reviewed literature establishes that contradictions emerge through social interaction between individuals and groups. This implies that the phenomenon under study fits with interpretive assumptions, that our knowledge of reality, including the domain of human action, is a social construction by human actors (Walsham 2006). The process of implementing an enterprise system in an organizational context is time consuming, and to understand how this process unfolds, a longitudinal study has been viewed appropriate. The interpretive research approach has therefore been guided by Pettigrew's (1995) advice on longitudinal research on change.

The ES implementation project lasted from 2002 to 2007 and has been studied from an outside observer viewpoint in three distinct periods. The time frame of the project and the research activities are outlined in Fig. 1. Above the time line are the main project phases, and below are the phases of research.

The data collection activities included a total of 23 interviews with 15 persons and about 840 pages of documents analysed. To get access to information-rich informants a partial "snowball" or chain-sampling of informants was used (Patton 1990). The

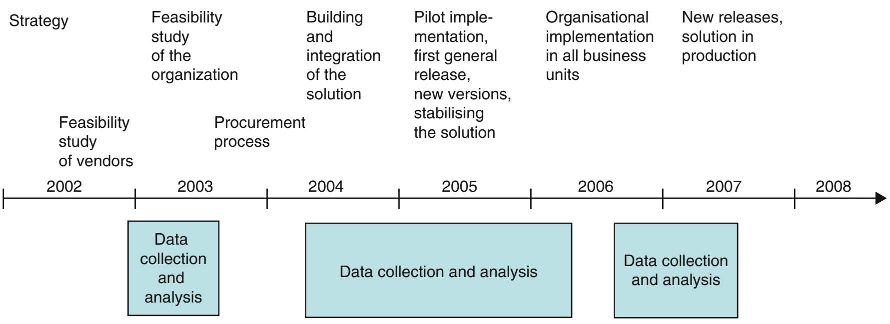


Fig. 1 The main project phases of the Statoil case above the timeline, together with research activities below

principal data collection method was in-depth, semi-structured interviews, combined with document analysis. To remain both focussed and open, a general interview guide approach was combined with an informal conversational interview, to “go with the flow” (Patton 1990). The interviews typically lasted 45 min. All interviews were audio-taped and transcribed.

The research heeded Pettigrew’s (1995) stance on how to study organizational change in context, namely that it requires multilevel analysis and processual analysis. Multilevel analysis was done at: (i) project level, (ii) group level with groups of users having diverging interests and (iii) individual level. The interviews were coded and a deductive analysis was based on the interview guides. Data reduction by tables and displays (Miles and Huberman 1994) were made. A visual mapping strategy (Langley 1999) was also used.

3.2 The Dialectic of Adaptation and the Appointment of a Special Role

In this case one main contradiction was found, termed the dialectic of adaptation. It is described in detail by Nordheim and Päivärinta (2006), and emerged between the features of the ES packages versus the organization-specific requirements addressed by the organization. The thesis in this contradiction was an ”out-of-the-box” implementation of commercial software. The antithesis was an implementation fulfilling Statoil’s requirements concerning solution integration and user experience. The contradiction may be phrased as follows: “we ought to implement commercial software as it is, out-of-the-box” vs. “no, we ought to implement the software so that we fulfill our requirements concerning solution integration and a simplified user experience”. The contradiction manifested as 42 critical ”issues” that jeopardized the entire project. They had to be solved before the project could continue (Nordheim and Päivärinta 2006).

As a result of and in a direct response to the dialectic of adaptation, a special role was established. Termed a corporate user representative, this role became an important way to deal with the dialectic of adaptation. A description of the corporate user representative role and how this role dealt with the contradictions is described in two previous publications (Nordheim 2008; Nordheim and Nielsen 2008).

4 Findings: Different Perspectives on Contradictions

In this case contradictions can be understood from the following perspectives:

- a pre-implementation perspective, in view of ES vendor challenges
- an early chartering phase perspective, by a focus on perceived customization needs
- a project phase perspective, as the contradictions manifested as critical issues

The understandings of contradictions from these perspectives are presented in the following.

4.1 Understanding Contradictions in View of ES Vendor Challenges

As pointed out in Sect. 2.2 a basic challenge that ES vendors face, was termed a dialectic of design (Nordheim 2007). This challenge is to find an optimal balance between stability and change, between commonality and variability. By comparing the contradictions involved in the dialectic of adaptation with the vendors' challenging dialectic of design, commonality and variability are underlying the issues in the contradictions.

Commonality was underlying the governing principle of using standard software "out-of-the-box". As the project manager during the most intense project phase stated:

> We were required to solve everything "out-of-the-box", as a general principle. As we unpacked the solution "out-of-the-box" and saw how it worked, we quickly found out that we had to do quite a lot. But there has always been a tension between the steering committee and the project concerning how far we should go with the local adaptation.

The limits of commonality can clearly be seen in the "out-of-the-box" pilot installation of the purchased products on Statoil's own ICT infrastructure. This resulted in a number of concerns, conceptualized as 191 "issues" identified for resolution during the implementation. The issues represented "weaknesses, challenges and uncertainties" in the solution, and 42 of the issues were categorized as highly critical (Nordheim and Päivärinta 2006).

Variability solved some of the issues, as they only required a simple configuration task. However, custom components were needed to supplement the limitations of commonality. The custom components were strictly prioritized, addressing solution integration and user experience. The limits of the built-in commonality and variability may also be illustrated by some statements by the corporate user representative on his tensions with different project groups:

> Is this really not possible to solve, with the products we bought?

> We have a problem here, what we bought turned out to be less flexible than we thought

> We have to change the scope, [we need to] do things differently, and the solution will be different from what we decided.

By analyzing the dialectic of adaptation in terms of the ES vendors' commonality/variability design challenges, these were clearly at the heart of the dialectic of adaptation. Commonality was largely expressed by the CIO's governance principle of using standard software "out-of-the-box". Variability related to solution integration and user experience was a particular challenge, and the case shows that variability was supplemented with custom components to achieve integration and user experience beyond the built-in variability of the software. The vendor challenges (Nordheim 2007) are therefore found to be underlying the dialectic of adaptation.

4.2 Exploring Manifestations of Contradictions Early in the Chartering Phase

An attempt to explore perceived customization needs was made early in the chartering phase (Nordheim and Päivärinta 2004). In retrospect this initial analysis can be viewed as an early attempt to understand possible latent contradictions that later surfaced in the ES implementation. Customization is certainly not recommended (e.g. Beatty and Williams 2006), but perceived needs for customization was here only used as a way to explore and anticipate possible contradictions early in the chartering phase. The study concluded that perceived needs for customization were mainly seen in the following areas: non-functional integration with existing software, simplification of user interfaces, and functional adaptation and simplification (Nordheim and Päivärinta 2004).

It turned out that exactly these customization needs later surfaced in contradictions. Integration and user experience later turned out to be key issues in the dialectic of adaptation (Nordheim and Päivärinta 2006). This shows that one could thus begin to understand important contradictions by focusing on perceived customization needs early in the chartering phase, before any contradictions emerged. With the benefit of hindsight, it can be seen that a focus on perceived customization needs early in the chartering phase was useful. This focus helped to uncover important issues that later emerged in the antithesis of the dialectic of adaptation.

4.3 Understanding Contradictions: Previous Findings from the Case

The case study has previously been analyzed in terms of a hybrid development approach involving three 'motors' of change (Nordheim and Päivärinta 2006). Although not explicitly focused on understanding contradictions, this study presents three important insights worth including here.

The first understanding of the dialectic of adaptation is its emergence. The dialectic of adaptation could be traced back to the project strategy. However, this dialectic was only latent in the beginning. It fully emerged in the project phase. The second understanding is that contradictions are only part of the picture. Viewed in terms of Van de Ven and Poole's (1995) four motors of change, a hybrid theoretical pattern of change was found best suited to describe the ES implementation. In other words, contradictions must in this case be understood as one of several drivers of change in the project organization. The third understanding of contradictions is that a stakeholder perspective is also important to understand the dialectic of adaptation.

Table 1 Perspectives for understanding contradictions in an ES implementation

Understanding	Description
Pre-implementation considerations: The vendors' dialectic of design	The dialectic of adaptation may be understood in view of the vendors' challenging dialectic of design, to find an optimal balance between commonality and variability. Commonality and limited variability was underlying the thesis of the dialectic of adaptation.
Early chartering phase exploration: Perceived customization needs for the ES	The dialectic of adaptation may early in the chartering phase be understood by exploring perceived customization needs. These needs included integration and user interface issues, and were later core issues in the dialectic of adaptation.
Project phase: emergence	The dialectic of adaptation emerged from being latent in the project strategy to surface as critical issues in the project phase.
Project phase: hybrid pattern	The dialectic of adaptation emerged in combination with the teleological and life cycle motors of change.
Project phase: stakeholders	The dialectic of adaptation occurred as representatives from different solution domains advocate their needs. Different user groups want to get "their" components prioritized first.

Diverging viewpoints on functionality surfaced within the steering group, and representatives from different domains advocated their needs. The emerging contradictions led to sharpened prioritizations of implementation issues and organizational learning among the stakeholders (Nordheim and Päivärinta 2006).

Table 1 summarizes the different understandings of contradictions obtained from this case study, and shows how the dialectic of adaptation may be understood from these perspectives.

5 Discussion

This paper summarizes understandings from one case. The research question was: How can we understand contradictions in enterprise system implementations? Given the dialectic of adaptation as a main contradiction, Table 1 summarizes different perspectives on this complex phenomenon.

The finding that the vendors' challenging dialectic of design was underlying the dialectic of adaptation, is a new contribution. It helps to understand why package-organization misalignments occur (Sia and Soh 2007). Of a more practical nature is the finding that an early focus on perceived customization needs surfaced issues in later contradictions. Issues in the dialectic of adaptation could be understood early in the chartering phase by focusing on perceived customization needs, and this is a new contribution.

Previous research on this case has established that contradictions emerged in the project phase, as part of a hybrid theoretical pattern of organizational change, and as the result of diverging viewpoints among user groups (Nordheim and Päivärinta 2006). Taken together, the findings from this case help us to understand some of the complexity underlying contradictions.

The point of understanding contradictions is to be better able to deal constructively with them. In this case the establishment of a corporate user representative role became an important way to deal with the dialectic of adaptation (Nordheim 2008; Nordheim and Nielsen 2008).

These findings should have relevance beyond this case, although there are some limitations on how the findings may be generalized: (i) the case was a combination of collaboration and information management technologies, (ii) the time of study is mainly the chartering and project phases, with only a few interviews early in the shakedown phase, (iii) the informants mainly represent the project perspective, and (iv) the focus is to a large extent limited to the dialectic of adaptation. However, the understandings of the dialectic of adaptation are related to the enterprise-wide nature of these systems, and should therefore be applicable to other enterprise systems.

Although the different perspectives presented here contribute to our understanding of contradictions in enterprise system implementations, the research question is still relevant. There may be other perspectives for understanding contradictions than the ones presented here, and future research could well focus on other contexts and later phases of ES implementation.

A conscious strategy of looking for contradictions and pursuing a constructive synthesis could help manage large scale ES projects. An early focus on perceived customization needs may as in this case help to uncover potential future contradictions in the project. One should also expect diverging viewpoints among user groups in the project phase, and one may anticipate these by appointing a carefully selected corporate user representative, as done in this case.

To conclude, two new understandings have been presented: (i) by considering the larger context of vendor challenges, the dialectic of adaptation can be viewed as manifestations of the vendors' dialectic of design, and (ii) by considering limits of variability in an early chartering phase, an early focus on customization needs helped to surface critical issues early. This paper also contributes by putting together a larger picture of how contradictions in an ES implementation can be understood. Understanding contradictions involves several dimensions and an openness to these could help identify and deal with contradictions in future ES projects.

References

Allen JP (2005). Value conflicts in enterprise systems. Inf Technol People 18(1):33–49

Beatty RC, Williams CD (2006) ERP II: Best practices for successfully implementing an ERP upgrade. Commun ACM 49(3):105–109

Besson P, Rowe F (2001) ERP project dynamics and enacted dialogue: perceived understanding, perceived leeway, and the nature of task-related conflicts. SIGMIS Data 32(4):47–66

Boonstra A (2006) Interpreting an ERP-implementation project from a stakeholder perspective. Int J Project Manag 24:38–52

Bühne S, Lauenroth K, Pohl K (2005) Modelling requirements variability across product lines. 13th IEEE International Conference on Requirements Engineering, Paris, France, IEEE Computer Society, pp 41–50

Davenport TH (1998) Putting the enterprise into the enterprise system. Harvard Business Rev 76(4):122–131

Israel J (1979) The language of dialectics and the dialectics of language. Munksgaard, Copenhagen

Langley A (1999) Strategies for theorizing from process data. Acad Manag Rev 24(4):691–712

Leishman DA (1999) Solution customization. IBM Syst J 38(1):76–97

Markus ML (1997) The qualitative difference in IS research and practice. Proceedings of the IFIP TC 8 8.2 International Conference on Information Systems and Qualitative Research, Philadelphia, pp 11–27

Miles MB, Huberman AM (1994) Qualitative data analysis: an expanded sourcebook. Sage, Thousand Oaks, CA

Munkvold BE, Tvedte B (2003) Implementing a portfolio of collaboration technologies in Statoil. In: Munkvold BE (ed) Implementing collaboration technologies in industry. Springer, London, pp 81–107

Nordheim S (2007) Towards a dialectic understanding of enterprise systems – vendor challenges and contradictory rhetoric. In: Magyar G et al (eds) Advances of information systems development: new methods and practice for the networked society, vol 1, Springer, New York, pp 11–22

Nordheim S (2008) Corporate user representatives and the dialectics of enterprise systems: a quest for social actors with political skill. Proceedings of the 41st Hawaii International Conference on System Sciences, Big Island, Hawaii. IEEE (CD-ROM)

Nordheim S, Nielsen PA (2008) Enterprise system implementations: organizational influence processes for corporate user representatives. Proceedings of the 16th European Conference on Information Systems, Galway, Ireland. (CD-ROM)

Nordheim S, Päivärinta T (2004) Customization of enterprise content management systems: an exploratory case study. Proceedings of the 37th Hawaii International Conference of System Sciences, Big Island, Hawaii. IEEE (CD-ROM)

Nordheim S, Päivärinta T (2006) Implementing enterprise content management: from evolution through strategy to contradictions out-of-the-box. Eur J Inf Syst 15:648–662

Päivärinta T, Munkvold BE (2005) Enterprise content management: An integrated perspective on information management. In: Sprague JR RH (ed) Proceedings of the 38th Hawaii International Conference on System Sciences, CD-ROM, IEEE, Los Alamitos

Patton MQ (1990) Qualitative evaluation and research methods. Sage, Newbury Park, CA

Pettigrew AM (1995) Longitudinal field research on change. In: Huber GP, Van de Ven AH (eds) Longitudinal field research methods: studying processes of organizational change. Sage Publications, Thousand Oaks, CA, pp 91–125

Robey D, Ross JW, Boudreau M-C (2002) Learning to implement enterprise systems: An exploratory study of the dialectics of change. J Manag Inf Syst 19(1): 17–46

Sabherwal R, Newman M (2003) Persistence and change in system development: a dialectical view. J Inf Technol 18(2): 69–92

Seddon PB (2005) Are ERP systems a source of competitive advantage? Strategic Change 14:283–293

Sedera D, Gable GG, Chan T (2004) Measuring enterprise systems success: The importance of a multiple stakeholder perspective. Proceedings of the 13th European Conference on Information Systems, Turku, Finland

Sia SK, Soh C (2007) An assessment of package–organisation misalignment: Institutional and ontological structures. Eur J Inf Syst 16:568–583

Smith HA, McKeen JD (2003) Developments in practice viii: enterprise content management. Commun AIS 11(33):1–26

Soh C, Sia SK et al (2003) Misalignments in ERP implementation: a dialectic perspective. Int J Hum Comput Interact 16(1):81–100

Soh C, Sia SK (2005) The challenges of implementing vanilla versions of enterprise systems. MIS Quart Executive 4(3):373–384

Van de Ven AH, Poole MS (1995) Explaining development and change in organizations. Acad Manag Rev 20(3):510–540

Walsham G (2006). Doing interpretive research. European Journal of Information Systems, 15: 320–330.

Ward J, Hemingway C, Daniel E (2005) A framework for addressing the organisational issues of enterprise systems implementation. J Strategic Inf Syst 14: 97–119

Wei H-L, Wang ETG, Pei-Hung J (2005) Understanding misalignment and cascading change of ERP implementation: a stage view of process analysis. Eur J Inf Syst 14(4):324–334

Perceived Quality of Cloud Based Applications for Collaborative Writing

Tihomir Orehovački

Abstract More recently, Internet has experienced a complete makeover and has developed into an always-connected and device-independent environment. Under the influence of new technological trends, including Web 2.0, service-oriented architecture (SOA), software as a service (SaaS), and cloud computing, there has been a shift in the approach to the development of Web applications. Static Web pages that used to be designed for passive recipients of information have been re-placed with dynamic applications that enable participants to actively contribute in creating knowledge repositories. Owing to their features, collaborative editors, as examples of SaaS, have a potential to be used for both educational and professional purposes. This chapter presents the results of our research into the perceived quality of collaborative editors in which two complementary methods were used: the retrospective thinking aloud (RTA) method and a questionnaire. Through analysis of collected data, advantages and disadvantages of collaborative editors were obtained. In addition, the guidelines for their improvement and further development of similar types of applications are proposed.

1 Introduction

A decade ago, the Internet was composed of static Web pages that were used by organizations to reach potential customers mainly for e-commerce and marketing purposes. Besides, they used the intranet to distribute business information among employees. Only persons with a substantial ICT knowledge were able to create Web sites and publish information online while users were just passive consumers and information recipients. A breakpoint in the classical approach to the use and development of Web sites occurred in September 2005, when Tim O'Reilly introduced the concept of Web 2.0 [14]. The dominant model of static Web sites was replaced by the paradigm of the social Web aimed at publishing facilitated by dynamic and

T. Orehovački (✉)
Faculty of Organization and Informatics, University of Zagreb, Varaždin, Croatia
e-mail: tihomir.orehovacki@foi.hr

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_46,

flexible Web applications that are interactive and more responsive to user actions. The idea at the core of Web 2.0 is harnessing collective intelligence, according to which users, through interaction and collaboration, actively participate in content repository creation, modification and sharing. The newly created repository has an added value because it contains the knowledge and experience of a large number of people and can be used for educational and professional purposes. In addition to the paradigm of the social Web, new models have appeared that along with the aforementioned paradigm form a complementary solution of a complex architecture.

The first among them is the service oriented architecture (SOA). SOA promotes the use of loosely related services in order to provide maximum flexibility in a business-interoperable and technology-independent way [13]. Cloud computing is an emerging paradigm that aims at delivering both software applications as services over the Internet and hardware infrastructure in data centers that provide those services [12]. The services themselves have long been referred to as Software as a Service (SaaS). SaaS is a type of cloud computing that enables the delivery of an application that is located in the infrastructure of the cloud to a number of users through a Web browser. Consequently, instead of investing in new servers and licensed software, users can hire an application and use it in the form of a service. An example of SaaS are collaborative editors, cloud based Web services that represent a replacement for standard “office” applications. The main advantages of collaborative editors are efficient formatting and editing of the content and simultaneous work on the same document. Moreover, such editors contain information about who created a document and when, which simplifies the management of document versions and their ownership. Finally, this type of applications includes a number of add-ons such as live chat, live markup and annotation etc., which significantly improves and facilitates the process of collaboration. Thus, collaborative editors have a potential to be used for both professional and educational purposes. However, collaborative editors as well as other cloud based applications present many challenges for system and application developers. Some of the major concerns when shifting to cloud based applications are related to security, privacy, trust, scalability, reliability and quality of service [3]. Implementing the principles of Web Engineering is proposed as a possible response to these challenges [6], especially the principle of quality assessment during the development process of SaaS. In order to determine to what extent the quality of a collaborative editor corresponds to users’ needs, a research was conducted with students who used such applications to perform educational e-activities. This paper presents the results of the research in the form of advantages and disadvantages of the existing collaborative editors, relative importance of quality attributes and recommendations for the development of cloud based applications.

2 Background of the Research

The assumption of the research is that the success of Web 2.0 applications depends on the largest index of quality among alternative solutions. The first problem that arises is the question of quality attributes that influence the success of Web 2.0 applications.

Namely, in order to create a high quality Web 2.0 application, developers must identify the key attributes that influence user behavior related to the use of Web 2.0 applications and Web site re-visit. Irani [9] has shown that a gap between users and developers of Web-based information systems exists and has not been resolved with a number of quality evaluation models [1, 4, 5, 18], frameworks [2] and standards [10, 11]. Furthermore, Zhang and von Dran [19] pointed out that the existing approaches to Web applications evaluation do not have a firm theoretical basis or empirical validation. The aforementioned issues account for a lack of guidelines to be used when developing Web 2.0 applications. Finally, recent studies [8, 16, 17] have shown that traditional approaches to Web pages quality assessment are not only unsuitable for a new generation of Web applications, but are also in conflict with their basic concepts and principles. Therefore, based on our literature review, a conceptual model of Web 2.0 applications quality evaluation was developed [15]. Within a model, quality attributes were classified into six basic categories: system quality (SYQ), service quality (SEQ), information quality (INQ), performance (PFM), effort (EFO) and acceptability (ACP).

System Quality includes six attributes (navigability (NVG), consistency (CNS), aesthetic (AES), familiarity (FML), customizability (CUS) and security (SCR), and refers to the functionality features of the Web 2.0 application that is used to perform an e-activity. The category that measures the quality of interaction between a Web 2.0 application and the user is called *Service Quality*, and consists of eight attributes: helpfulness (HLP), availability (AVL), interactivity (ITR), error prevention (ERP), reliability (RLB), recoverability (RCV), responsiveness (RSP) and feedback (FDB). *Information Quality* can be considered from two aspects. One is the quality of information content which is the result of using a Web 2.0 application, and the other is the quality of the information content which is located on the Web 2.0 application. In both cases, the information content should match the following quality attributes: correctness, coverage, credibility, timeliness and value-added. For the purpose of this research, the quality of information is perceived only within the first aspect. *Performance* refers to the quality of e-activities execution by using Web 2.0 application interface elements. This category consists of only three attributes: effectiveness (EFE), usefulness (UFL) and efficiency (EFI). *Effort*, as the name suggests, refers to the amount of the expected effort when using a Web 2.0 application, and includes the following attributes: minimal action or physical effort (MAC), minimal memory load or mental effort (MEL), accessibility (ACS), controllability (CTR), ease of use (EOU), learnability (LRN), memorability (MRB) and understandability (UND). To facilitate data collection in this study, two theoretically separated attributes, that is, minimal action and minimal memory load, are logically combined into a single attribute that is named physical and mental effort (PME). The last category, *acceptability* consists of attributes that directly contribute to the success of a Web 2.0 application, including playfulness (PLY), satisfaction (STF) and loyalty (LOY). A detailed description of all these attributes can be found in [15].

Apart from being distributed into six basic categories, quality attributes can be classified into two main groups, according to the evaluation methods used, that is, into subjective and objective attributes. Subjective attributes are related to the perception of the Web application quality arising from the experience of its use and are primarily estimated by using inquiry methods, most common among which are questionnaires. On the other hand, objective attributes refer to technical aspects of the functioning and use of a Web application and are most often measured by using testing methods (e.g. thinking aloud or automatic tools for usability evaluation). In our study, subjective and objective attributes were evaluated from the perceptive aspect of the user employing both inquiry and testing methods.

3 Research Method

In order to gather qualitative and quantitative data, two complementary methods were used: retrospective thinking aloud (RTA) and the questionnaire. The experiment consisted of subjective quality assessment of four different cloud based applications (Google Docs, Zoho Notebook, iNetWord and Helipad) that can help in the implementation of collaborative educational e-activities. The first two applications were chosen because they are the ones most commonly cited in the context of collaborative writing (instead of the Zoho Notebook, a Zoho Writer is mentioned). The other two applications were intentionally included in the research due to their features. The reason why we selected four different Web applications with a similar purpose was to engage study participants in solving similar tasks in several different ways and thus enable them to evaluate which collaborative editor is most appropriate for a given type of educational e-activities. In addition, respondents were able to compare the used editors and therefore better perceive their shortcomings.

During the study, which lasted an entire semester, students used collaborative editors to complete four e-activities. In order to better assess their quality and determine their advantages and disadvantages, students used a different collaborative editor for each e-activity. Educational e-activities consisted of several parts. Firstly, students took notes in class by using a collaborative editor, and when the class was over, they had to expand their notes with complementary literature. In this way, all the students worked together to create a common repository of knowledge which they could use in their preparation for the final exam. At the end of each e-activity, students expressed their critical opinions, including the advantages and disadvantages of the collaborative editor used while performing the e-activity. In order to avoid the negative effects of using classical thinking aloud concerning e-activities, the RTA method was used [7]. RTA allows users to complete the e-activity first, and then verbalize their experiences of working with a Web application.

At the end of the semester, students completed a questionnaire assessing the quality of all the four Web 2.0 applications. Developing our own questionnaire was primarily motivated by the fact that all the existing ones were designed to estimate the quality of static Web sites or general software products and do not include all the important quality dimensions that we attempted to cover in this study. Participants provided their replies to questionnaire statements in the computer lab, that is, in controlled

Table 1 Data obtained from the questionnaire

	Helipad ($n = 167$)		iNetWord ($n = 171$)		Zoho Notebook ($n = 165$)		Google Docs ($n = 168$)	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
SYQ	3.20	1.095	3.69	1.119	3.92	0.903	4.47	0.714
SEQ	3.50	1.113	3.54	1.182	3.82	0.994	4.24	0.827
INQ	2.83	1.030	3.19	1.036	3.53	0.927	4.05	0.863
PFM	3.00	1.290	3.57	1.199	3.80	1.109	4.39	0.807
EFO	3.32	1.271	3.80	1.130	3.99	1.005	4.56	0.669
ACP	2.34	1.191	2.97	1.226	3.30	1.186	4.16	1.000

conditions. For each quality attribute a default statement was created, and students expressed their agreement or disagreement with a given statement concerning each collaborative editor using a five-point Likert scale. Combining the questionnaire with RTA, complementary data of the perceived quality of the collaborative editors used was gathered. By using the questionnaire, we collected students' subjective assessment of quality attributes for each of the collaborative editors. On the other hand, the data obtained from RTA provided us with more detailed information regarding the advantages and disadvantages of cloud based applications for collaborative writing in terms of quality attributes. The research results are explained in more detail in the following section.

4 Findings and Discussion

The study involved 174 undergraduate students of Information Science. However, as the amount of usable data on the respondents' part varies from one editor to another, the numbers listed in the tables in this section refer only to respondents whose data were properly submitted and could therefore be considered representative. At the time when the experiment was conducted, participants were in the second year of study. Among the respondents, 70% percent were male and 30% female. All of them used Web 2.0 applications on a regular basis (70.11% do so twice a day or more often). While performing e-activities, respondents were able to use an operating system and a Web browser of their own choice. The summary of data on major quality categories collected from the questionnaire is presented in Table 1, while the data gathered from RTA is given in Table 2.

It should be noted that the results in Table 2 are presented for each attribute of the five basic quality categories separately, while the data for the category information quality is presented cumulatively.

According to the results obtained from the questionnaire, only the availability and security attributes were assigned high values. Namely, almost all respondents reported that all the four editors were available whenever they were accessed and that the algorithms implemented ensured user privacy and security of their data. The mean value ranged between 4.92 and 4.56 for the security attribute and between 4.73

Table 2 Data gathered from the retrospective thinking aloud method

	Helipad ($n = 132$)		iNetWord ($n = 134$)		Zoho Notebook ($n = 130$)		Google Docs ($n = 133$)	
	Pros	Cons	Pros	Cons	Pros	Cons	Pros	Cons
NVG	5	23	17	3	25	5	21	1
CNS								
AES	17	15	20	30	38	3	28	7
FML	11	23	41	2	20	4	55	
CUS	6	32	24	10	27	7	46	5
SCR	2		11	7				
HLP	6	6	1	1	2			
AVL	7	1	16	7	21		33	1
ITR	23	6	34	15	28	7	39	2
ERP		1		27	1	3		2
RLB	30	5	21	37	17	7	49	3
RCV		1	2	11	1			
RSP	1		3	6	1	2		1
FDB				6				
INQ	4	4	1	4	14	6	4	4
EFE	5	4	39	19	42	19	63	12
UFL	24	105	37	18	63	22	108	3
EFI	10	9	12	13	4	3	15	2
PME	3	29	2		4	2	2	1
ACS			3	2				
CTR	1	2	1	4	2	29	4	3
EOU	53	26	80	3	68	4	103	1
LRN	3	8	11	3	9	9	13	
MRB								
UND	2	5	3	1	5	7	7	2
PLY	3	1	11		2		7	
STF	32	56	57	32	87	16	107	3
LOY	1	11	8	3	5	2	11	
CMP[a]	8	1	5	22	4	6	10	
INT[a]	2	1	5	3	1	7	11	3
SLB[a]	2	4	13	8	15	15	39	5

[a]Attributes were not included in the original questionnaire.

and 4.18 for Web application availability. A detailed data analysis is further provided separately for each collaborative editor.

4.1 Helipad

Data obtained from both methods of quality evaluation indicate that Helipad (*pad.helicoid.net)* is the least appropriate solution for the implementation of collaborative educational e-activities; in comparison to the other three editors, it received the lowest ratings. A more detailed analysis shows that the greatest shortcoming of this application is its inability to adapt to the characteristics of e-activities and personal preferences of users (mean = 2.86). Specifically, participants reported that this editor has a very few options for text processing, which prevented them from emphasizing some parts of the text which they thought were important. Therefore, it is not surprising that the perceived quality of information content that can be created by using this tool is relatively low (mean = 2.83).

Another major problem related to this application is that it bears little similarity to an alternative desktop or Web application (mean = 2.79). As a matter of fact, Helipad is more similar to a wiki system or a small compiler than a text editor. Most of the symbols that the other text editors use for basic mathematical operations are used for text formatting in Helipad (e.g. * for bold, - for italic). Furthermore, when the students wanted to add snippets of C ++ code to their notes, the result of using this application was content with highly inconsistent formatting features. It is interesting to note that this somewhat different approach to text formatting caused a high degree of dissatisfaction among students, especially if we consider that they are students of Information Science and should therefore be ready to embrace new approaches that resemble programming. Moreover, the TAB key in this editor does not have the same purpose during text writing as is the case with other applications that were used in the study. Another disadvantage of this application is a lack of a classic menu or icons to perform basic text formatting operations (mean = 2.86). This drawback was the main reason why a large number of users were rather confused before they realized that Helipad offers a "Formatting Guide" as a replacement for menus and icons. Yet another shortcoming of this editor perceived by users is a lack of interface functionalities that refers to the effectiveness and efficiency attributes. Namely, users found the tool too simple for the execution of collaborative activities, which suggests that using the tool for such purposes may not be recommendable. As can be seen in Table 2, a very large number of users reported that it would be much easier to perform the same e-activity with an alternative desktop or Web application. Finally, the application does not have an option of storing files on a server and if users want to import an image into a document, they must use another online service and then, within the Helipad, insert a link to the image. Because of all the mentioned limitations of this editor, participants have concluded that it needs to undergo major changes before it can be used for demanding e-activities and become a good replacement for standard editors. Therefore it is not surprising that a relatively small number of users intend to use this editor in the future or recommend it to their colleagues (mean = 1.74).

Naturally, Helipad has its advantages, the most important among which is its reliability. According to the data obtained by both methods, a very small number of users reported a negative experience related to errors or interruptions when using it. However, due to its numerous flaws, this very important quality attribute can pass almost unnoticed.

4.2 iNetWord

iNetWord (www.inetword.com) is the third-ranked collaborative editor according to the assessment in our research. Among its advantages we should emphasize a minimum amount of effort required for the use and performance of e-activities (mean = 3.80). In addition, this editor is very similar to the existing desktop applications with the same purpose. However, its greatest advantage is a set of add-ons (e.g. built-in spellchecker and FTP access to data) that the other three applications do not have and which encourages participants to use it in the future.

On the other hand, this editor has some serious shortcomings, including reliability, error prevention and recoverability (mean = 3.33, 3.02 and 3.25, respectively). When saving a created document on a server, either manually or automatically, the application would return an incomprehensible error after which the data were irretrievably lost. In this case, users had to start performing their activity from scratch. Another significant oversight of this application is definitely its design which is old-fashioned and does not include pastel and vivid colors that characterize the new generation of Web applications (mean = 3.46). In addition, the toolbar is overcrowded with options which reduce the accessibility of icons that are placed on it. Finally, it should be noted that a major drawback of this editor reported by a large number of users is its incompatibility with most Web browsers (it only worked in Internet Explorer), especially with Google Chrome, which is a specialized thin client for working with cloud based applications. Altogether, the editor received a passing grade (mean = 2.97), but if we consider the intention of future use only, the assessment ratio is still very low (mean = 2.13).

4.3 Zoho Notebook

According to its overall score, Zoho Notebook (notebook.zoho.com) was the second-ranked editor. It should be noted that Zoho Cloud includes another application called ZohoWriter that may be even more appropriate for educational e-activities. However, as one of the major goals of this study was to identify drawbacks in the development of different types of Web applications whose primary purpose is collaborative writing, we have deliberately chosen Notebook, which is used for storing notes. Zoho Notebook is an application of a fairly good quality. This particularly applies to the total effort that needs to be invested in the use of this application (mean = 3.99). Users believe that the use of this editor does not require any prior knowledge and that it is very easy to navigate through it. Furthermore, it has an intuitive interface and good design (mean = 3.90). Among the reasons why someone would continue to use this application, respondents identified the possibility of synchronous communication via chat and organization of notes in the form of a notebook, which contributes to a higher visibility of created documents. However, its narrow specialization in the management of notes was perceived as quite a disadvantage. Namely, users had problems when they tried to insert a picture inside a document, which resulted in a huge number of window sliders and poor viewability of the document. Nonetheless, the relatively large number of users believe that by using this Web application they can make a lot of useful and similar e-activities, which accounts for high levels of satisfaction and application acceptability (mean = 3.30).

4.4 Google Docs

Among all the four editors that were used in the study, Google Docs (docs.google.com) received the highest ratings. From the long list of its advantages, we must emphasize its ease of use (mean = 4.67), very simple, clear and intuitive interface (mean = 4.51) and great similarity with the related desktop applications (mean = 4.47). However, this application also has some drawbacks, as indicated by the data obtained from the RTA method. According to some participants, Google Docs still does not have enough options to fully replace alternative desktop applications; it works much slower with larger amounts of text, and does not have a feature for organizing a document into pages. In addition, the objections of individual users are related to the interface design that is considered to be too simple, and the fact that the first sentence is automatically treated as a title, without the possibility to be modified by the user. However, this editor gained the highest ratings in all the six categories of quality attributes. Such ratings may suggest that, owing to its positive characteristics, this application is an example that others should follow in the development of similar types of cloud based applications.

5 Recommendations, Conclusions and Future Work

The success of a Web 2.0 application depends on loyal rather than casual users. Therefore, the main challenge for any developer is to create a Web application that will meet the users' needs and encourage them to re-visit a particular Web site. However, research has shown that a gap still exists between how the developers envisage quality attributes of a Web application and the users' actual needs. The result is a large number of unusable Web applications that fail to achieve their two primary objectives: satisfying the users' needs and increasing the developers' profits. In order to reduce the mentioned gap to a minimum, a research that proposed a set of attributes for assessing the quality of Web 2.0 applications was conducted [15]. The main objective of this paper was to check the suitability of the set of quality attributes proposed in the cited paper on a sample of collaborative editors. The second goal was to complement research results obtained by the questionnaire with data yielded from the RTA method. Finally, based on the list of gathered advantages and disadvantages, yet another goal of this study was to offer guidelines and recommendations for further development of collaborative editors. The following passages comprise a list of the attributes most frequently mentioned by the respondents in the context of advantages and disadvantages of used collaborative tools along with their description. This list should serve as a set of recommendations and guidelines for developers in further development of cloud based collaborative applications.

Satisfaction refers to participants' responses regarding their feelings when using Web application (i.e., whether they are happy or frustrated when they have to perform an e-activity with its interface functionalities). In order to attract as many new users and retain the existing ones, a Web application must meet the users' expectations and needs in terms of its appearance and features as closely as possible. *Usefulness* depends on the features and functionalities offered by a Web application and illustrates whether its interface enables users to solve real problems in an acceptable way. A user will certainly continue to use the Web application if it contains all the features which ensure that a given task can be accomplished in a better way than with a desktop or Web alternative. *Ease of use* is the degree to which users believe that using a Web application will be free from difficulties and that they will manage to accomplish a specific task without the help of any kind. If a Web application has a number of other qualities, but the execution of e-activities is considerably more difficult than with a similar desktop or Web application, users will still undoubtedly turn to alternatives. *Effectiveness* is the capability of a Web application to enable users to achieve specific e-activities with completeness and accuracy. *Reliability* refers to the amount of errors and interruptions when using a Web application. In order to enable users to work smoothly, we need to implement mechanisms that will reduce the occurrence of errors and interruptions to a minimum. In cases when errors and interruptions are unavoidable, it is necessary to protect information from loss, implement mechanisms for recovery from interruptions and inform users about the cause of the error and its type. *Aesthetic* refers to the interface design and visual appearance of the content of a Web application. The interface design is the first thing that users notice in a Web application. For example, if the colors are not in contrast and the text is not readable, users will not even begin to use interface elements but will browse for some harmonious alternative instead. *Customizability* refers to the possibility of adapting Web applications to personal preferences, and cultural and local characteristics of users (e.g. spell check or support for diacritical marks) or the characteristics of the specified task (e.g. use of templates). *Familiarity* is the degree to which the design, layout and use of interface elements are similar to previously used applications. Since users are unwilling to accept applications that are much different from those they use on a daily basis, it is very important to design and organize interface elements and thus make them more intuitive for the user. *Interactivity* is a quality attribute according to which a Web application creates a feeling of use of a desktop application. A Web application should contain features such as auto save of changes on a document, integrated chat for communication and synchronous operation on a specified task, the possibility of publishing and sharing documents with colleagues, etc. *Navigability* is a quality attribute according to which interface elements must be well organized so that the user can find them quickly and easily. In addition, a Web interface should offer alternative navigation mechanisms (e.g. a search engine), which enhances the use of its functionalities and resources.

The results of our research showed that users placed a greater emphasis on subjective than objective quality attributes. Furthermore, it should be noted that while thinking aloud, users did not mention the attributes *consistency* and *memorability* even once. On the other hand, three quite important attributes were often mentioned and they may be added to future versions of our questionnaire. The first attribute is *scalability* (SLB), an extent to which a Web application enables a smooth execution of e-activities regardless of the number of users simultaneously accessing its interface functionalities or the scope of the task. Next attribute is *compatibility* (CMP), according to which interface functionalities of a Web application work equally well in all Web browsers. The last attribute is *interoperability* (INT), which refers to the ability of a Web application to interact and exchange files with other desktop or Web systems.

Our future work will be focused on the implementation of Web 2.0 applications to other types of educational e-activities. Moreover, we will test an improved new questionnaire in the area of e-commerce. In addition, to improve the applicability of the proposed set of quality attributes and assessment methodology, research will be extended to include domain experts. Finally, by using automatic tools for usability evaluation, we will gather data concerning the objective attributes that will surely enrich and complement the results presented in this paper.

References

1. Abramowicz W, Hofman R, Suryn W, Zyskowski D (2008) SQuaRE based Web Services Quality Model. Proceedings of the International MultiConference of Engineers and Computer Scientists, IAENG, March 19–21, Hong Kong, pp 827–835
2. Ahn T, Ryu S, Han I (2007) The impact of Web quality and playfulness on user acceptance of online retailing. Inf Manag 44(3):263–275
3. Buyya R, Pandey S, Vecchiola C (2009) Cloudbus toolkit for market-oriented cloud computing. Lect Notes Comput Sci 5931:24–44
4. Calero C, Ruiz J, Piattini M (2005) Classifying web metrics using the web quality model. Online Inf Rev 29(3):227–248
5. Delone WH, McLean ER (2003) The DeLone and McLean model of information systems success: a ten-year update. J Manag Inf Syst 19(4):9–30
6. Deshpande Y, Murugesan S, Ginige A, Hansen S, Schwabe D, Gaedke M, White B (2002) Web engineering. J Web Eng 1(1):3–17
7. Guan Z, Lee S, Cuddihy E, Ramey J (2006) The validity of the stimulated retrospective think-aloud method as measured by eye tracking. Proceedings of Conference on Human Factors in Computing Systems, April 22–27, Quebec, Canada, pp 1253–1262
8. Hart J, Ridley C, Taher F, Sas C, Dix A (2008) Exploring the facebook experience: a new approach to usability. Proceedings of the 5th Nordic conference on Human–Computer Interaction: Building Bridges; ACM; October 20–22, Lund, Sweden, pp 471–474
9. Irani Z (2002) Information systems evaluation: navigating through the problem domain. Inf Manag 40(1):11–24
10. ISO/IEC (1998) ISO 9241-11:1998—Ergonomic requirements for office work with visual display terminals (VDTs)—Part 11: Guidance on usability. International Organization for Standardization, Geneva
11. ISO/IEC (2001) ISO/IEC 9126-1: 2001—Software engineering—Product quality—Part 1: Quality model. International Organization for Standardization, Geneva

12. Marinos A, Briscoe G (2009) Community cloud computing. Lect Notes Comput Sci 5931:472–484
13. Marks EA, Bell M (2006) Service-oriented architecture (SOA): a planning and implementation guide for business and technology. business and technolog Wiley, New York
14. O'Reilly T (2005) What is Web 2.0: design patterns and business models for the next generation of software, http://oreilly.com/web2/archive/what-is-web-20.html [1/19/2010]
15. Orehovački T (2010) Proposal for a set of quality attributes relevant for Web 2.0 application success. Proceedings of the 32nd International Conference on Information Technology Interfaces, IEEE, June 21–24, Cavtat, Croatia, pp 319–326
16. Silva PA, Dix A (2007) Usability—not as we know it! Proceedings of the 21st British HCI Group Annual Conference on HCI 2007: People and Computers XXI HCI... but not as we know it—Volume 2, ACM, September 3–7, University of Lancaster, UK, pp 103–106
17. Thompson A-J, Kemp EA (2009) Web 2.0: extending the framework for heuristic evaluation. Proceedings of the 10th International Conference NZ Chapter of the ACM's Special Interest Group on Human–Computer Interaction, ACM; July 6–7, Auckland, New Zealand, pp 29–36
18. Venkatesh V, Morris MG, Davis GB, Davis FD (2003) User acceptance of information technology: toward a unified view. MIS Quart 27(3):425–478
19. Zhang P, von Dran GM (2000) Satisfiers and dissatisfiers: a two-factor model for website design and evaluation. J Am Soc Inf Sci 51(14):1253–1268

From Evidence-Based to Knowledge-Based Healthcare: A Task-Based Knowledge Management Approach

Hamidreza Pousti, Henry Linger and Frada Burstein

Abstract The healthcare is a knowledge-based profession where ideally decisions are based on formal evidence. But the challenges faced by healthcare decision makers are to apply these generic and population level evidence to the specific situation of an individual patient. In this paper we argue for a knowledge-based approach to medical decision-making. Such an approach is grounded in general population level interventions, based on the evidence that informs and is specialized to the specific context of a particular patient. We propose task-based knowledge management (TbKM) as a theoretical construct to implement the knowledge-based approach to medical decision-making.

1 Introduction

Success of a society is always linked to its health and wellbeing. Health and health systems are fundamentally important from both economic and social perspectives as a healthy population can underpin strong economic growth and community wellbeing and prosperity (Suhrcke et al. 2006). Development of an efficient healthcare system presents a prime concern for every society since there are considerable challenges and issues to deal with. Many of those challenges are concerned with responding to the increasing complications of the healthcare environment such as growing cases of chronic diseases or the shortage of vital skills in the industry.

Healthcare is widely recognized as a knowledge intensive community that relies on knowledge creation and sharing and within the last 15 years evidence-based medicine (EBM) has become an essential part of healthcare delivery (Haynes 2006). In EBM, medical decisions heavily rely on the results attained from the gold standard of medical studies known as double-blind randomized controlled trials or RCT (Elstein

H. Pousti · H. Linger · F. Burstein (✉)
Faculty of Information Technology, Monash University, Melbourne, Australia
e-mail: hamid.pousti@infotech.monash.edu.au

H. Linger
e-mail: henry.linger@infotech.monash.edu.au

F. Burstein
e-mail: frada.burstein@infotech.monash.edu.au

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_47,

2004; Ncayiyana 2007). The main goal of RCT is to determine the efficacy of a specific medical intervention on a cohort who represents a broader population (Evans 2003). EBM also relies on monitoring new interventions that are universally applied to the population at large (Caulfield 2007). This cycle of RCTs and the extensive method of monitoring of EBM contribute to the generation of new knowledge about the treatment of a particular condition and importantly the limitations of a specific intervention.

Guyatt et al. (2004) argue that the limitations of EBM arises from the fact that many important medical decisions need to remain sensitive to patients' specific conditions, values and preferences. Sackett et al. (1996) believe that without careful consideration of patients' contexts, the risk of evidence–based medical decisions risk become oppressive since even excellent evidence may be inapplicable to or inappropriate for an individual patient. Woolf et al. (2005) and Corrao et al. (2008) concur with Sackett et al. (1996) and argue that the quality of clinical decisions can be improved through the development of effective knowledge management program, leading to a more effective healthcare system.

In this paper we argue for a knowledge-based approach to medical decision-making. Such an approach is grounded in EBM as a general, population level intervention that informs and is specialized to the specific situation of an individual patient. We propose task-based knowledge management (TbKM) as a theoretical construct to implement the knowledge-based approach to medical decision-making.

This study is organized as follow. In the next section the significance of healthcare systems and the challenges they face are discussed. This section sheds more light on the context and state of healthcare problems in Australia by looking into Australia healthcare system and the nature of this industry. The following section is concerned with how knowledge management addresses those issues followed by the penultimate section where we explore what solutions task-based knowledge management model can offer. We conclude with suggestions on further areas of investigation.

2 Healthcare System and its Challenges

Health systems generally face enormous challenges that are mainly concerned with the development and provision of a variety of services to meet their ever increasing consumer demand. Meeting these challenges will ensure that healthcare systems remain a reliable and effective driving force for social and economic development (Bloom et al. 2004). Therefore, it is imperative for healthcare systems to develop a mix of complex services that effectively respond to their stakeholders' requirements. On the other hand, the development of these complex services inevitably relies on scarce resources.

Healthcare systems usually have to deal with limited financial and human resources. These limitations can widen the gap between health consumer demands and the supply of the health services. As Lindholm et al. (2001) argue this gap can have unfavorable impacts on the quality of the healthcare services in long term may even have severe consequences on economic growth and social development. For example, the healthcare system is one of the largest and most complex industry sectors in Australia and in order to maintain its health outcomes, the Australian healthcare system has to carefully monitor and deal with two major sets of challenges that can affect its performance adversely (AHMAC 2008). The first set is mostly concerned with issues around the rapid growth of consumer demands (Armstrong et al. 2007). Although these are radical challenges, its discussion is beyond the scope of this article. The second set of challenges resides in the nature of the industry as a knowledge-based industry sector. In a healthcare system there is a broad range of knowledge-intensive communities, such as physicians, specialists, nurses, and patients who make the system dependent on knowledge and knowledge becomes a driver across the industry (Fox et al. 2005; Ghosh and Scott 2007).

Because of the high level of dependency on knowledge, the industry strive to find effective ways of dealing with the increasing necessity to capture, manage and apply the health related information and knowledge (National EHealth Strategy 2007). In fact the major players of the industry have come to the realization that properly addressing the knowledge challenges can pave the way to implement a reliable and sustained healthcare system that supports all aspects of healthcare from planning, to management and deliverying services.

To achieve this goal, it is crucial to maximize the efficacy and efficiency of the healthcare system. This can be achieved by deploying the knowledge residing with its workforce, as developing extensive information bases to support effective decisions and actions. However, Jadad and Enkin (2000) highlight that the new era of information poses significant challenges to the processes of knowledge creation, dissemination, and application in healthcare systems.

Technology has enabled enormous volumes of information to be generated and readily accessible. This trend has been extenuated by the Internet as well as many tools that allow the rapid and wide dissemination of information. A remarkable similar information revolution occurred during the 16th century when laboriously transcribed manuscripts were replaced with rapidly reproducible and portable printed books (Drucker 1999; Eisenstein 2005).

In fact, despite all the benefits of the new era of information revolution, many knowledge workers feel uncomfortable in fulfilling their tasks as they often have to try hard to keep up with the knowledge being generated in their field (Jadad et al. 2000). One common and notable issue is that with so much knowledge available, knowledge workers cannot possibly absorb it all. The exponential growth of formal research studies is generating new and rich sources of scientific information at an unprecedented rate (Anderson and Graham 1980; Benseman and Barham 2009). In the field of biomedical research for example, medical knowledge doubles in amount every 2 years and physicians need to know many of the half a million articles added each year to the biomedical literature (Wyatt 1991; Hanka and Fuka 2000; Davenport and Glaser 2002).

Today, one of the major challenges for knowledge workers is to narrow the gaps between what they could know and what they do know and between what they think they should do and what they really do. While not restricted to healthcare, all knowledge workers have the problem of staying on the top of the knowledge available in their field. However, unlike other fields, healthcare errors are life threatening.

The increasing gap between the creation of biomedical knowledge and the capacity of healthcare workers to acquire that knowledge may have significant impacts on the healthcare delivery process and it can undermine the quality of the care with some severe consequences. This is consistent with the finding of Davenport and Glaser (2002) that show more than a million injuries and as many as 98,000 deaths each year are attributable to medical errors and more than 5% of patients had adverse reaction to drugs while they were under medical care and nearly half of them were either serious life threatening or fatal. Some of these mistakes result from carelessness, but most occur because the clinicians must track a massive amount of complex information. Nichols et al. (2008) categorize medical errors into three major categories: lapse of memory or slip in attention due to tiredness and shift work; error of judgment due to misinterpretation of patient situation; and errors due to lack of knowledge and inaccessibility of right information resources. Two of these three major categories contributes to more than 50% of errors and can be directly related to the lack of knowledge or the ways information is organized and/or presented. In fact, in many cases information and knowledge presentation are ineffective and do not meet the specific requirements of healthcare service delivery and can have detrimental effect on medical decisions.

From medical decisions view, healthcare workers need fast and complete access to the right information that is consistent with the current medical status of their patients and is presented at the correct level of abstraction. The ability to correctly and efficiently process mass quantities of information are central to correct and timely decisions and to achieve this, it is imperative to consider the context of information in order to make appropriate judgment (Eichler et al. 2004; Chaudhry et al. 2006).

3 Knowledge Management in Healthcare

Healthcare decisions today need to be based on evidence and such a decision making process is an essential practice for all healthcare workers regardless of where they work and what sources of information they have access to. The effectiveness and reliability of the evidence in turn depends on the ability of healthcare workers to access relevant and current knowledge bases. This concept is valid for any healthcare decision maker irrespective of whether they work in an advanced and high-end hospitals in a developed country or working in remote areas of a developing country (Mouhouelo et al. 2006).

Medical decision making requirements are mainly concerned with the explicit aspect of knowledge that is captured and codified so as to provide a formal evidence base for the decision. As Al-Hawamdeh (2002) states there is wide agreement that such explicit knowledge can be considered information. However, knowledge by its very nature depends on other knowledge to build on and draws on individuals' personal construction of reality (Sveiby 1999). This type of knowledge is referred to as tacit knowledge that cannot be codified, but it is a dispensable part of all knowledge. Tacit knowledge is the most elusive way of knowing, but perhaps it is the most important way. "Know-how" is a knowledge created out of practice and collectively shared by workgroups and it is the part of tacit knowledge that can be explained or even codified (Teece 1998).

According to Benson and Standing (2004) the goal of knowledge management (KM) is to identify who has the relevant tacit knowledge and how others can identify, access and use that knowledge when they need it. On the other hand, explicit knowledge is also required to justify healthcare related decisions and ground it in the body of evidence. Knowledge management reflects a concern for developing a well-expressed and long-term plan for the intellectual asset of knowledge-intensive organizations that have to respond effectively to their dynamic, unpredictable, and complex environments. This situation fits well to the healthcare environment where the clinical decision making process is based on physicians' intuition and formal bodies of evidence (Wyatt 2001).

4 Task-Based Knowledge Management (TbKM) in Healthcare

For knowledge management to be effective and successful it must be a visible part of the work practice with outcomes that are supportive to organizations sustainability and performance. In the context of healthcare, Davenport and Glaser (2002) argue that successful and most promising KM initiatives are the ones that can "bake" and embed specialized knowledge into the jobs of highly skilled healthcare workers. This is consistent with task-base approach to KM (Burstein and Linger 2003) where the processes that create and exploit knowledge are clear and focus on tasks and their actors. In fact TbKM changes the focus of evidence-based clinical decisions from a target population to an individual patient by providing a framework to appropriately apply evidence in the patient's context.

As Burstein and Linger (2003) argue, TbKM represents a generic framework that can be applied in most knowledge work tasks. The aim of this framework is to acquire, represent, preserve and distribute knowledge, created as a result of performing a specific task (Burstein and Linger 2003). In this framework a "task" is defined as a substantially invariant activity with tangible outputs where "knowledge work" refers to the collection of knowledge based activities that constitute a task. We recognize evidence-based decisions-making as knowledge work since it is considered to be a rigorous knowledge intensive process with explicit outcomes (Straus and Haynes 2009; Sackett et al. 1996) as well as conceptual outcomes that includes knowledge creation and learning (Burstein and Linger 2003).

In task-based KM, the concept of community of practice (CoP) is central to the success of the KM process. It incorporates both individual and organizational perspectives into knowledge work activities. In general, a CoP can encourage innovation, collaboration and sharing of good practice by harnessing the power and knowledge of individuals into collaborating works and allows all opinions to be heard (Wenger et al. 2002; Watson and Harper 2008). However, in the task-based framework, CoP plays a significant role in defining the task as a socially constructed activity. In such a setup, individual actors who form the CoP, perform their task with the help of technological components called "knowledge work support system" and they are responsible for deciding what knowledge is shared and when it will be shared. Therefore, it is necessary to maintain the integrity of each actor's individual perspective while the knowledge is shared within CoP.

From task-based KM perspective, in evidence-based medical decision making, decision making is considered a task where actors collaborate with each other and share their knowledge to achieve a particular decision. Based on this framework, activities such as research for finding relevant evidence is considered as part of the knowledge work carried out by CoP member. This shifts the process of evidence based decision making from independent decisions made by individuals to a collaborative decision made by the group of healthcare workers involved in that specific instance of that task. As the result decision makers are able to develop a common language for understanding their task, share meanings and, document aspects of the implicit knowledge that they applied to that instance of the task. This is a process of 'inscription' (Latour 1986) which can reveal the knowledge that is necessary for a reliable decision making.

Task-based KM also gives healthcare workers the ability to specialize their knowledge to fit the particular situation of a task. This instantiation allows each healthcare worker to exercise judgment and use her implicit knowledge to assess the applicability of existing material or past instances of the task to meet the contingencies of the current situation. Instantiation can have significant impact on the process of decision making where the supporting knowledge for a particular decision is complex or incomplete and past experiences of the situation can significantly contribute to the accuracy of decision. The contribution is not only in applying past experiences but also in recognizing the differences of past instances so that an appropriate approach can be applied in this instance.

In task-based KM, healthcare workers are located in a three dimensional space of doing/thinking/communicating. Within this space CoP validates the created knowledge and sanctions interpretations and meanings of the created knowledge by authorizing actors to make decision. This ensures that decisions made by healthcare workers are consistent as they accomplish their work activities in accordance with the body of knowledge sanctioned by the CoP.

Although task-based approach to knowledge management can make significant impacts on evidence-based decisions making in healthcare, implementing such a model as a Knowledge Management System (KMS) for professionals such as physicians is a challenging process. In many situations blending separate professional areas into a collaborative knowledge management process may be resisted as it may impact the influences of various contributors. In healthcare, physicians enjoy a high level of autonomy in making their decisions and they make decisions largely independent of other healthcare professionals. Findings from a study (Burnett et al. 2005) on a KMS implementation program in a healthcare centre shows that KM can change the whole doctor–patient relationship. In reality KM is a shift in organizational culture and adopting a new paradigm of organizational relationship. It is influenced by how individual knowledge workers regard their knowledge community and the role they should play. This is in harmony with Alavi and Leidner (1999) notion of KM as a social and cultural process that is in alliance with its other organizational and technological aspects.

As a socio-cultural and socio-technical phenomenon, KM in healthcare needs to rely on the contribution of a range of different professionals with different areas of expertise. In knowledge-intensive communities such as healthcare that rely on services, innovations, and knowledge sharing, the focus needs to be shifted from merely individuals' knowledge to the underlying values that are communicated and adopted by its members. In fact, KM processes should be recognized as cultural values and become part of the mainstream activities. In this context the aim of KM is to create harmony, consensus and cohesion among the all contributors (Debowski 2006). Therefore, KM in healthcare should be seen as a long term commitments that can change the culture of healthcare to a more collaborative and proactive community. By achieving this goal KM can influence retention and productivity (Sheridan 2002) by constructing a strong organizational culture around its activities and ensuring stable and predictable values that are sustained over a long period (Smith and Rupp 2002; Jones et al. 2003).

5 Conclusion

The research presented in this paper supports the notion that better healthcare systems can be beneficial for economic outcomes and social development. However healthcare systems are under pressure from their both internal and external environmental demands. Healthcare, is a highly knowledge-based industry that relies on evidence-based decisions. However it needs to embrace appropriate approaches to knowledge management in order to effectively apply population based evidence to the particular situation of individual patients. In this paper we argued that TbKM can provide a framework for such medical decision making by incorporating evidence with a patient's condition and values. In this way TbKM improves the decision making process and contributes to the quality of healthcare and builds a strong collaborative culture between healthcare workers involved in those decisions. However, we acknowledge that the process of implementing a knowledge management system is a shift in organizational culture in healthcare as its processes can change the dynamics of the relations between healthcare actors and impact on the social structures of organization.

References

AHMAC (2008) National E-Health strategy: summary. In AHMAC. Australian Health Ministers' Conference. Victorian Department of Human Services

Alavi M, Leidner DE (1999) Knowledge management systems: issues, challenges, and benefits. Commun AIS 1(2es):1

Al-Hawamdeh S (2002) Knowledge management: re-thinking management and facing the challenge of managing tacit knowledge. Inf Res 8(1):1–15

Anderson J, Graham A (1980) A problem in medical education: is there an information overload? Med Educ 14(1):4–7

Armstrong BK et al (2007) Challenges in health and health care for Australia. Med J Aust 187(9):485–489

Benseman J, Barham P (2009) Coping with information overload: the development and evaluation of a journal review service. Med Educ 18(6):446–447

Benson S, Standing C (2004) Effective knowledge management: knowledge, thinking and the personal—corporate knowledge nexus problem. Springer Netherlands 3(2):227–238

Bloom DE et al (2004) The effect of health on economic growth: a production function approach. World Dev 32(1):1–13

Burnett S et al (2005) Knowledge support for interdisciplinary models of healthcare delivery: a study of knowledge needs and roles in managed clinical networks. Health Inform J 11(2):146–160

Burstein F, Linger H (2003) Supporting post-Fordist work practices: a knowledge management framework for supporting knowledge work. IT People 16(3):289–305

Caulfield J (2007) Clinical trials—shedding light the evidence. Pharmacist 26(6):450–454

Chaudhry B et al (2006) Systematic review: impact of health information technology on quality, efficiency, and costs of medical care. Ann Int Med 144(10):742–752

Corrao S et al (2008) Evidence-based knowledge management: an approach to effectively promote good health-care decision-making in the Information Era. IEM 4:99–106

Davenport T, Glaser J (2002) Just-in-time delivery comes to knowledge management. Harvard Business Rev 80(7):3–9

Debowski D (2006) Knowledge management, 1st edn. Wiley, Australia

Drucker P (1999) Beyond the information revolution. The Atlantic Monthly (October) 284:47–57

Eichler H et al (2004) Use of cost-effectiveness analysis in health-care resource allocation decision-making: how are cost-effectiveness thresholds expected to emerge? Value Health 7(5):518–528

Eisenstein E (2005) The printing revolution in early modern Europe. Cambridge Uniersity Press, Cambridge

Elstein AS (2004) On the origins and development of evidence-based medicine and medical decision making. Inflamm Res 53(S2):184–189

Evans D (2003) Hierarchy of evidence: a framework for ranking evidence evaluating healthcare interventions. J Clin Nurs 17:77–84

Fox NJ et al (2005) The 'expert patient': empowerment or medical dominance? The case of weight loss, pharmaceutical drugs and the Internet. Soc Sci Med 60(6):1299–1309

Ghosh B, Scott J (2007) Effective knowledge management systems for a clinical nursing setting. Inf Syst Manag 24:73–84

Guyatt G, Cook D, Haynes RB (2004) Evidence based medicine has come a long way: the second decade will be as exciting as the first. BMJ Oct 30:990–991

Hanka R, Fuka K (2000) Information overload and "just-in-time" knowledge. Electronic Library 18(4):279–285

Haynes B (2006) Of studies, syntheses, synopses, summaries, and systems: the "5S" evolution of information services for evidence-based healthcare decisions. Evidence Based Med 11(6):162–164

Jadad A et al (2000) The Internet and evidence-based decision-making: a needed synergy for efficient knowledge management in health care. CMAJ 162(3):362–365

Jadad A, Enkin M (2000) The new alchemy: transmuting information into knowledge in an electronic age. JAMC 162(13):1826–1828

Jones N, Hershel R, Moesel D (2003) Using "knowledge champions" to facilitate knowledge management. J Knowl Manag 7(1):49–63

Latour B (1986) Visualization and cognition: thinking with eyes and hands. Knowledge Soc 6:1–40

Lindholm C, Burstro M, Diderichsen F (2001) Does chronic illness cause adverse social and economic consequences among Swedes? Scand J Public Health 29:63–70

Mouhouelo P, Okessi A, Kabore M (2006) Where there is no internet: delivering health information via the blue trunk libraries. PLoS Med 3(3):e77
National E-Health Strategy (2008). http://www.health.gov.au/internet/main/publishing.nsf/Content/National+Ehealth+Strategy. Accessed 18 Apr 2010
Ncayiyana DJ (2007) Evidence-based medicine is not all randomised controlled trials and systematic reviews. South Afr Med J (January) 97:310
Nichols P et al (2008) Learning from error: identifying contributory causes of medication errors in an Australian hospital. Med J Aust 188(5):276–279
Sackett D et al (1996) Evidence-based medicine: what it is and what it isn't. BMJ 312(7023):71–72
Sheridan JE (2002) Organizational culture and employee retention. Acad Manag J 35(5):1036–1056
Smith A, Rupp W (2002) Communication and loyalty among knowledge workers: a resource of the firm theory view. J Knowl Manag 6(3):250–261
Straus S, Haynes RB (2009) Managing evidence-based knowledge: the need for reliable, relevant and readable resources. CMAJ 180(9):942–945
Suhrcke M et al (2006) The contribution of health to the economy in the European Union. Roy Inst Publ Health 120:994–1001
Sveiby KW (1999) The tacit and explicit nature of knowledge. In the knowledge management yearbook 1999–2000. Butterworth-Heinemann, Bostan, MA, pp 18–36
Teece DJ (1998) Capturing value from knowledge assets: the new economy, markets for know-how, and intangible assets. Calif Manag Rev 40(3):55–79
Watson K, Harper C (2008) Supporting knowledge creation—using Wikis for Group Collaboration. Educ Center Appl Res 2008(3)
Wenger E, McDermott RA, Snyder W (2002) Cultivating communities of practice. Harvard Business Press, Cambridge
Woolf SH et al (2005) Promoting informed choice: transforming health care to dispense knowledge for decision making. Ann Int Med 143(4):293–300
Wyatt J (1991) Use and sources of medical knowledge. Lancet 338(Nov):1368–1373
Wyatt J (2001) Management of explicit and tacit knowledge. J Royal Soc Med 94(1):6–9

Agile Support and Maintenance of IT Services

Jaroslav Prochazka

Abstract In this paper, we deal with agile and lean support and maintenance of IT services. Agile approaches are considered as software development approaches mostly, but the usage of agile principles and techniques also brings a lot of benefits to the support and maintenance. Common support and maintenance standards, recommendations and methods are process oriented and omit human aspect. Paper takes a critical viewpoint of process-orientation. We define agile and lean support and maintenance principles and control framework for improvement and implementation of these principles in IT services. The described approach was piloted in 13 IT services. Achievements are enclosed as verification of practical results and benefits.

1 Introduction

Agile approaches have been widely accepted as software development approaches in IT industry during the last ten years. Agile techniques are becoming very popular because of their effectiveness and efficiency. We can name Daily meetings, Pair Programming, Test Driven Development or Continuous Integration as the most known. We can see the importance of human aspect rather than the detailed process descriptions in agile and lean approaches. In the support and maintenance area, there still remains process oriented approach, e.g. ITIL [1], ISO 12207 Maintenance sub-process [2, 3] or IEEE 1219 Standard for Software Maintenance. Software development approaches have evolved during past decades from ad hoc chaotic activities to iterative and incremental (agile) processes. We can see the same evolution in the support and maintenance approaches but shifted several years backwards. Now we still live in a process period. Why do we not consider lessons learnt in agile development approaches and evolve support and maintenance approaches the same way?

J. Prochazka (✉)
Tieto/University of Ostrava, Ostrava, Czech Republic
e-mail: jaroslav.prochazka@osu.cz

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_48,

Support and maintenance is one of the longest phases in the IT service lifecycle, thus the overall costs spent are very significant. The most money is spent on maintaining and supporting steady state, less than half of IT budget is spent on innovations [4, 5]. Poor support and maintenance approach can significantly increase the costs and what's more, it can decrease the product quality and affect motivation of the support and maintenance teams. We introduce particular problems and proofs of support and maintenance that we've dealt with and propose a solution framework consisting of agile and lean support and maintenance principles, measuring framework and implementation approach.

1.1 Support and Maintenance

Terminology recapitulation follows to baseline reader's understanding and terminology. The IEEE definition of software maintenance is following [3]:

> Software maintenance is the process of modifying a software system or component after delivery to correct faults, improve performances or other attributes, or adapt to a changed environment.

This definition has been further discussed resulting in ISO standard for software life cycle processes [2]. It concludes software maintenance as a discipline that can start early in the process and is all about fixing defects. Software maintenance considers following categories [2, 6]: *corrective* (reactive modification), *adaptive* (implementing changing requirements), and *perfective* (improvements).

IT service support is understood as activities described in ITIL [1], namely Incident, Event and Problem management processes. These processes are integrated in function called Service Desk. Service Desk serves as a single point of contact (SPOC) for business users and defines the window to IT department or IT vendor. As the last one we state IT service definition [1]:

> A service is a means of delivering value to clients by facilitating outcomes clients want to achieve without the ownership of specific costs and risks.

There exist a lot of formal recommendations or standards in software support and maintenance area. These are namely ISO 12207 Maintenance sub-process [2, 3], IEEE 1219 Standard for Software Maintenance [7], ITIL framework [1], ISO 20000 [8] or MANTEMA method [9].

1.2 Agile Software Development and Maintenance

This section summarizes agile software development principles and state of the art of agile maintenance. Agile development comprises several methods or frameworks, namely Scrum [10], Extreme Programming [11], Lean Software Development [12], Feature Driven Development, Crystal methods and others. People working on these methods met in 2001 and formed so called Agile Manifesto [13] defining principles and values common for these methods. In Agile Manifesto, you can see stressed (1) communication and cooperation among all people (business and IT; within IT teams); (2) focus on continuous delivery of valuable software; (3) working software as the primary measure of progress; and (4) change tolerance understood as an advantage for the customer, not as an impediment. This can be taken as a short introduction to agile methods. Agile approaches occur as a response to the waterfall and spiral model problems.

Agile and research communities apply agile and lean methods in maintenance area after the year 2000 as well. Their application is described in various numbers of experience reports or research papers. There exist two tracks of applicability of agile and lean in maintenance: namely method track (e.g. applicability of XP [14] or Lean software development [12, 15]) and practice level track (e.g. [16, 17]). It should be also mentioned that maintenance is also incorporated in agile software development methods. In agile, teams continuously deliver releases and as part of development they also correct defects incoming from operational environment. But this is not true if following traditional methods of software development and maintenance.

2 Initial State and Problem Discussion

We start this section with problem definition. Brooks identifies complexity, conformity, changeability and invisibility [18] as four essential difficulties of software and those are valid also in maintenance. Canfora and Cimitile [19] add to mentioned ones also program comprehension, impact analysis and regression testing as maintenance specific issues. Those issues are generally valid. But we have observed also issues related to current maintenance approaches, it is namely:

1. Process orientation (prescribed phases, detailed activities) omitting human aspect by assuming the same result with different people, it has two consequences:
    a. detailed procedures constraint creativity and proactive behavior;
    b. comprehensive documentation is the main form of collaboration and knowledge sharing and storage;
2. Measuring process performance (KPIs) rather than final result (business value—how much money will customer organization make or save by delivering new feature that can be measured as new revenue, revenue increase or operational efficiency [20]; Total Cost of Ownership; decreased maintenance costs);
3. Often lacking practical implementation approach.

Project or service consists of three different aspects: People, Process and Technology (tools, technologies, frameworks used for automation of work in the process). Current standards (ISO, IEEE, ITIL) focus on process aspect mostly. Agile approaches focus on people instead; the process, technology, tools and level of documentation depends on people's skills and experience. A similar people-centric approach is also verified by more than a 50-year-experience in Toyota [15].

Listed issues make maintenance more expensive and end user satisfaction and maintenance teams' motivation lower. Typical are recurring incidents and defects because of lack of architectural knowledge in the maintenance teams, fire fighting and reinventing the wheel (poor knowledge sharing). Let us bring empirical evidence confirming these claims:

- the cost spent on maintenance is more than 50% of IT Budget, e.g. [4, 5];
- Standish group Chaos research showing low success rate of traditional methods [21] and customer researches complaining to missing proactive behavior and innovativeness and alignment with customer business goals, e.g. [22];
- overhead cost spent on documentation, sharing and reuse problems [14, 19, 23];
- low maintenance motivation discussed e.g. by Barry Boehm in [24];
- experience reports, e.g. [14, 19, 22, 23, 25] to name a few;

Also our corporation Tieto (with its near 17,000 people seated around the whole world) had the same experience. The initial state of our approach (before we applied defined agile and lean support and maintenance principles) was standard reactive Incident Management process for handling and processing incidents, service requests and changes based on ISO 20000 [8] and IEEE 1219 Standard [7]. There was no time and motivation of teams for proactive work (in ITIL [1] known as Problem Management). Enhancements were implemented following role driven phase model leading to a long defect fixing period or low end user satisfaction at the end. Standard hierarchical management and decision making supported by corporate Project Management process based on IPMA recommendations. Communication in this model was often tool-driven (issue tracking tool) or email-driven. Maintainers did not discuss much with end users, only for short time before and after delivery. Disconnected was also the technical operations team with the one responsible for software side of a service. People were following the stated processes with no invention and proactive behavior.

3 Solution: Support and Maintenance Control Framework

Governance and control model is needed in organization to measure benefits of the services and to know their status. Not every organization measures and knows the status of its deliveries (projects or services). If they measure and know the status, they do not know what actions to take when measures breach the threshold values. Most organizations simply do not have a framework to know what changes to make and what to target for improvements [26].

The main contribution of this paper is defined *Control Framework of Agile and Lean Support and Maintenance* helping organizations to tackle problems mentioned in chapter "Initial State and Problem Discussion", to connect IT services with business goals, and to mitigate the risks. By the control framework is meant the whole agile and lean support and maintenance approach consisting of the (1) principles we defined based on existing research and empirical experience, as discussed in chapter

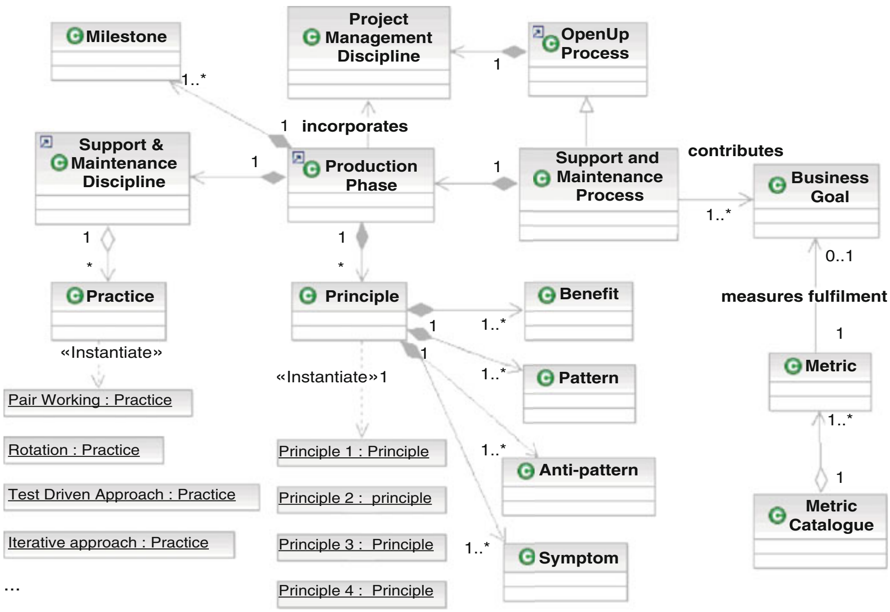


Fig. 1 Structure of the agile and lean support and maintenance control framework: production phase is the container for agile and lean principles and practices. Resulting process being built from the framework contributes to technical team innovativeness and achievements of customer's business goals and its contribution is measured

"Initial State and Problem Discussion", (2) control framework with measuring principle and (3) implementation approach. The structure of defined framework describes Fig. 1.

The basic container of Agile and Lean Support and Maintenance is the Production phase extending OpenUP software development framework [27]. The purpose of the Production phase is to operate, support and maintain IT service in an effective and efficient manner as well as to satisfy the customer by implemented changes and defect fixes. But Production phase can be connected to any development model or process. Objectives for the phase have been defined (as is common for OpenUP phases):

- Operate the system/service and support the users.
- Implement changes and fix defects (corrective change); improve the software design, code, functions (perfective change)—refactor the code to keep the code base maintainable; keep up with new technologies (to remain competitive).

The Production phase starts when the first release to the production environment is done, and ends when the Product Retirement milestone is reached. As a part of Production phase we defined four basic principles of Agile and Lean Support and Maintenance (see next chapter). The Production phase also contains disciplines performed to deliver the value. The Support and Maintenance discipline is a container for agile and lean practices and techniques that are basic building blocks of the final Support and Maintenance process. Practices can be added independently and incrementally according to the need, not all are needed in all IT service operations and maintenance. What more, we can measure their contribution to the business goals.

3.1 Our Principles of Agile and Lean Support and Maintenance

The basic building blocks of the Production phase are four principles of Agile and Lean Support and Maintenance. These principles shall be understood as an extension to Agile Manifesto values [13]. Each principle consists of benefit, pattern (the approach how to achieve the benefits), anti-pattern (what to avoid), symptoms (an indicator for the team whether the principle is followed or not), and practices (set of relevant agile and lean practices and techniques supporting implementation of the principle in the practice). Let me discuss in more details defined principles of agile and lean support and maintenance of complex IT services.

3.1.1 Principle (1): More Discipline, Less Bureaucracy

Benefit: enterprise ability to react to changes; commitment to follow the rules, processes because of team's contribution to their definition; motivated people; fun at work.

Pattern: define basic rules with cooperation of the team members (bottom-up approach); do not detail activities to the last detail, let the room for creativity and automate basic rules; perform regular retrospective and change the rules if needed; flat hierarchical structure (less managers, more leaders in teams with decision power).

Anti-pattern: define activities and processes to the last detail on the management level (without workers' involvement) not to allow the room for mistakes, omissions; leave the rules without changes for years (people need to get used to) and add new rules continuously; do not leave any decision power on the team but give them all the responsibilities.

Symptoms: every change approval takes time with involvement of many management levels; there exist enterprise business system (processes) not followed by people; a lot of regular useless reports required by managers; process oriented metrics not showing the real delivery status (e.g. plan fulfillment, utilization); faking reports; unproductive meetings; decreasing technical team motivation.

Agile Practices Supporting This Principle: iterative approach, kanban, retrospectives, rotation, visualization, continuous integration.

3.1.2 Principle (2): Internal and Cross Team Cooperation

Benefit: fulfilled needs of the users; informed users of IT service; meeting the business goals using effective IT service; contribution to the business value creation; fun at work.

Pattern: share organizational business goals with the all employees; map business goals to the IT (projects, services); define metrics from the business goals (not only from process efficiency) perspective; rotate the people in the development and maintenance teams and also in the support teams; regular demonstration of new/changed functionality.

Anti-pattern: defy the people from the business side, they do not know programming and understand technologies and us, IT gurus; carry off information and explanation of management decisions, people do not need to know them; establish different teams for development and maintenance of the services.

Symptoms: role oriented way of working; knowledge and information transfer via documents; unknown business goal in the teams; sub-optimization; unknown service or delivery status; 90% done syndrome; IT understood as the cost.

Agile Practices Supporting This Principle: visualization, business scenarios, pair working, rotation, on-job learning, retrospective, fight with ambiguity.

3.1.3 Principle (3): Proactivity

Benefit: the customer satisfied with operations, support and maintenance; the business is supported by appropriate IT services (based on proposals and implemented improvements); new business opportunity, new business orders from the customer.

Pattern: understand the business and propose future IT evolution according to the business scenarios; analyze trends, simulate and propose possible solutions to the customer; prototype to evaluate and measure your proposals.

Anti-pattern: only solve incidents and minor problems when occur (you are paid for this); do not ask for business evolution and changes, new offices and services planned by a customer; do not propose solutions to problems because customer will not pay them.

Symptoms: unfamiliarity of customer's business processes; daily activities are mostly fire fighting (reactive behavior); no time to do things right, to implement improvements (extend test coverage, refactor, reorganize components, investigate root causes); vendor does not solve (propose solutions to) customer's daily problems.

Agile Practices Supporting This Principle: iterative approach, retrospective, rotation, visualization, business scenarios, on-job learning, fight with ambiguity.

3.1.4 Principle (4): Risk Driven Approach

Typical statement about operations and maintenance is: “There are not many risks there”. “We have mitigated the most of them in the development phase.” This statement is true if we followed risk driven approach like Rational Unified Process [28] or OpenUP [27], but in reality we experience integration problems and cumulated regression defects after the regular release [4, 23, 25]. The symptoms of unsolved risks remain and the following new risk contributors appear during operations: changing market needs (changes in IT services); new functionality and defect fixes; changes of laws and ministry office edicts; existing IT service technologies (architecture, external interfaces, use of new technologies and integration of new services), and team members (their knowledge, fluctuation, motivation) and other resources, contracts, dependencies, and constraints of services.

Benefit: mitigate the level of risks to acceptable level (to avoid financial losses, delays, quality problems) and plan the service/project according the existing constraints.

Pattern: fact identification; risk identification and given action proposal including assignee and deadline; given actions are performed as real work (integration, development, prototyping) instead of a detailed theoretical analysis or monitoring.

Anti-pattern: ignore surroundings and changes; solve problems and risks when they appear; always conduct deep and detailed analysis before real implementation, integration, simulation, tests; only monitor the appearance of risk.

Symptoms: reactively driven service (only acting when problems occur); surprises during operations resulting in financial losses, delays or poor quality; facts interpreted as risks; no time to do things right.

Agile Practices Supporting This Principle: iterative approach, retrospective, business scenarios.

3.2 Practices for Agile Support and Maintenance

Four principles define the basic behavior, the core of the future support and maintenance process. Techniques and practices are concrete steps how to implement a principle in the real life. Agile practices are implemented incrementally according to our need aiming to achieve and contribute to enterprise business goals and to cover a gap in the principles. It is neither mandatory to implement all practices, nor to do it at the same time as a big bang. The result of the described approach (Production Phase, principles, practices) is an effective (not bureaucratic) maintenance process supporting objectives of delivery (project or IT service).

The basic set of the agile and lean practices and techniques in Agile and Lean Support and Maintenance according to our experience is following:

- iterative approach/kanban—to plan and manage workload and people, demonstrate, test and deliver value frequently, gather lessons learnt and improve the process;
- pair working (instant code review, better quality within shorter time [29, 30])—not ad hoc but planned one, for several hours per day used for critical incident solving, problem root cause investigating, change request design or code review;
- rotation—to avoid reinventing the wheel and spread the knowledge over the maintenance and support teams (rotate people between the development and maintenance teams and also between L2 and L3 Service Desk levels);
- test driven approach (unit testing)—to get quick feedback and to achieve higher product quality when changing the code, fixing a defect, implementing new functionality; the following patterns are applied:
  - when fixing a defect—unit tests are implemented for the class/package/module being fixed,
  - when implementing a change—modified class/package/module is covered by unit tests,
  - when implementing a new feature—it is covered by unit tests as a part of the implementation,
- refactoring—to incrementally clean and improve the code structure and to improve the architecture to acquire maintainable code base; a unit test suite is needed to avoid regressions;
- daily meetings—to synchronize in the team, share solutions, share service status and identify problems early;
- on-job learning—to gain knowledge and experience quickly;
- business scenarios simulation—to adapt IT services to evolving business and be ready for cost efficient future solutions using state-of-the-art technologies;
- retrospective—to gather lessons learnt and act upon if weak areas identified;
- continuous integration—to get quick feedback and improve the quality of the code/service;
- defensive programming—to avoid common defects, follow the conventions and have readable code for easier maintenance;
- planning poker—to plan the content of iteration more precisely and collectively and to have fun at work;
- visualization—general technique for uncovering the problems or trends using e.g. burn down chart (progress), kanban dashboard (uncover small problems and dependencies, or too much work in progress), value stream analysis;
- fight with ambiguity—techniques (memory heuristic, keyword technique, fuzzy sets and fuzzy logic) to deal with different languages of business and IT people.

The set of agile practices and techniques is not limited to the listed one. These practices have been proven in our context and IT services. To get a detailed description of agile and lean practices in software development context, you can read original

Table 1 Achievement of business objective is supported by standard operational objectives [26, 37]. Based on IT service context, we measure the operational objective with the relevant measures (our control framework). To improve operational objectives we implement appropriate agile and lean practice(s). Example shows practices, operational objectives and measures helping to achieve 10% turnover increase in one year (business objective)

Business objective	Operational objectives	Operational objective measures	Set of relevant agile practices
Increase the turnover per one customer by 10% until the end of this year	Improve the quality	Number of critical defects Ratio open/closed incidents Trend of repeating incidents Test coverage in %	Test driven approach Iterative approach Continuous integration Pair working
	Increase productivity	Team velocity Mean time to repair (MTTR)	Retrospective Rotation Pair working On-job learning

author's books [10–12, 27, 28]. As for empirical studies proving the effectiveness and efficiency of agile, lean practices in software development, we can name e.g. [16, 29, 30, 35] and their usage [31], partially [12] and [32]. As for studies focused on agile maintenance, we can name, e.g. [14, 16, 17, 33].

3.3 Our Implementation Approach

Patterns mentioned in the context of each principle guide us how to implement the principle in the practice (in IT service). Of course, exact steps depend on the service specific organization structure, team knowledge and skills, technology, customer and vendor's culture, phase of the lifecycle and contract constraints. We introduce an implementation approach verified in real implementations. Costs and risks of this approach are mentioned in chapter "Preferences of Business Process Models: Interpretative Analysis on Spontaneously Common Symbols". Let's have more detailed description of the steps:

1. *Initial analysis* consists of interviews with different team members, stakeholders and managers, way of working workshops (mapping the waste using Value Stream Analysis [12]), artifacts walkthrough and business goal identification. Based on this, we map the reality and identify root causes of the problems using Current Reality Tree (CRT) [34] or Kaizen workshop [15].
2. Next step is to choose and connect business and operational goals with *relevant agile and lean practices* (as an example see Table 1).
3. Basis for the *roadmap definition* is a set of chosen prioritized practices connected with business goals (from the step 2), identified risks and also important issues perceived by the team or by the customer. The final step is preparation of an

Table 2 Achievements in two IT services when we applied defined approach

	Service X—forest	Service Y—telecommunication
Short background	Data warehousing and ETL processing (200 workflows) supporting billing business process; 4 people in service, 5 maintaining Informatica platform (mixed teams in Czech and Finland)	L2 and L3 support and maintenance team, also developing small change requests; invoice processing in batches (XML and various scripts); 4 people in Czech, 2 people in Sweden; millions of invoices processed weekly
Perceived problems (by customer and teams)	1. Unknown SLA targets 2. Recurring incidents 3. Critical incidents caused by platform and database	1. Invoices processed in batches, failure required restart of the whole batch 2. Increasing (geometrically) processing time causing problems with invoicing 3. Very low motivation of the team
Achievements after applying our approach	1. Identified dependencies and SLA times thanks to early piloting in transition period 2. 75% decrease of incidents in 5 months, changed nature of incidents (to low impact and priority ones) 3. Doubled end user satisfaction	1. New 160 000 EUR business for our company (development projects for rewriting to real time processing) 2. Customer's daily critical business problem solved 3. Increased team motivation (team having also development projects)

implementation roadmap to support incremental adoption of the agile practices. Two types of metrics are set. Adoption level metrics (measuring the level of adoption of the practice) and business metrics (S.M.A.R.T. [36]).

4. Mentor help the team to *implement the chosen practices on daily basis* (with cooperation of local team change agent) in the form of hands-on support using coaching, facilitating and/or leading meetings, workshops, retrospectives. We also serve as SPOC to answer all the questions of the team members.
5. The steps are measured and regularly evaluated and are repeated until the desired business objectives are achieved.

3.4 Evaluation of the Approach in IT Services

The applicability of defined approach and framework has been verified in 13 real IT services from different business domains (telecommunication, banking, government, forest and energy and university) with European customers inside and outside our corporation. Due to limited space of this paper, we will demonstrate achieved results in two IT services, remaining ones have achieved similar sort of improvements (new development because of proactive improvement proposals, decreased incidents, quick transition and knowledge sharing, cost saving with higher quality—harder **SLA**) (Table 2).

3.4.1 General Survey of the Teams

Our approach is proven not only by hard measured data but also by the survey done among the team members and managers (42 managers, architects, leaders, testers have answered) where we applied approach described in this paper. The survey was focused on tangible and measurable aspects of the service but also on soft aspects (end user satisfaction, team motivation, benefit of our solution). The results of this survey prove our statement related to the applicability of people driven approach in support and maintenance (to name one: 66.7% of respondents answered that our approach helped them completely to achieve business objectives). The result shows that application of our approach helps to achieve delivery business and IT objectives.

4 Conclusions

We have discussed problems of traditional process-oriented approaches in support and maintenance and proposed our approach based on defined principles of agile and lean support and maintenance consisting of control framework with measures and connected practices. As an empirical evaluation we applied the approach in 13 IT services in distributed environment. Based on the results, we can state that described approach can mitigate problems of traditional methods (low innovativeness, creativity and motivation; process, not value oriented measures; quality problems).

We should also stress the costs and risks of this approach, not to name only benefits. The most risky part is misunderstanding of the principles. Agile and Lean Support and Maintenance is not a prescribed detailed process, appropriate process is built up from practices and differs in every team (different focus, practices to be implemented). The critical success factor mentioned by the team members and managers is hands on support by mentor (skilled and experienced person with agile approaches) helping to identify the root causes and implement principles and proper practices. The costs and overheads connected with the implementation are very low. The necessary activities performed beyond daily work are minimal (extra synchronization and follow-up meetings). The amount of overheads in all services was around 5% (around 2 h per week for sync meetings and preparation for workshops) of the team effort. The reason of this is the nature of the implementation approach we follow (hands-on). No new tools need to be introduced, changes are implemented incrementally and the whole mentoring is done on-job.

References

1. OGC (2007) The official introduction to the ITIL service lifecycle, version 3. TSO, London
2. ISO 12207:2008 (2008) Standard for software lifecycle processes
3. IEEE Std. 610.12 (1990) Standard glossary of software engineering terminology. IEEE, Computer Society Press, Los Alamitos
4. Hopkins R, Jenkins K (2008) Eating the IT elephant. IBM Press, Upper Saddle River
5. Boehm B (1981) Software engineering economics. Prentice Hall, Englewood Cliffs

6. ISO 14764:2006 (2006) Standard for software lifecycle processes—maintenance
7. IEEE 1219 (1998) Standard for software maintenance
8. ISO 20000 (2007) Standard for service management—Part 1, Part 2
9. Polo M, Piattini M, Ruiz F, Calero C (1999) MANTEMA: a software maintenance methodology based on the ISO/IEC12207 standard. In: Fourth IEEE international symposium and forum on software engineering standards, pp 76–81
10. Schwaber K (2004) Agile project management with scrum. Microsoft Press, Redmond
11. Beck K (1999) Extreme programming explained. Embrace change. Addison-Wesley, New York
12. Poppendieck M (2006) Implementing lean software development. Addison-Wesley, New York
13. Agile Manifesto (2001) http://agilemanifesto.org. Accessed 21 Jan 2010
14. de Souza SC et al (2007) Which documentation for software maintenance? J Braz Comput Soc 13(2):31–44
15. Liker J (2003) The Toyota way. McGraw-Hill, New York
16. Rudzki J, Hammouda I, Mikkola T (2009) Agile experiences in software service company. In: SEAA conference, IEEE Computer Society, pp 224–228
17. Svensson H, Host M (2005) Introducing an agile process in software maintenance and evolution organization. In Proceedings of the 9th European conference on software maintenance and reengineering, pp 256–264
18. Brooks FP Jr (1987) No silver bullet. IEEE Comput 20(4):10–19
19. Canfora G, Cimitile A (1995) Software maintenance. In Proceedings of the 7th international conference software engineering and knowledge engineering, pp 478–486
20. Bills M (2004) ROI: bad practice, poor results. Cutter IT J 17(8):10–15
21. Standish Group (2009) Chaos research 1999–2009
22. Martorelli B (2010) Application outsourcing clients are satisfied, but want more. Forrester (Published on April 21)
23. Lientz BP, Swanson EB (1981) Problems in application software maintenance. CACM 24(11):763–769
24. Selby R (2007) Software engineering: Barry W. Boehm's lifetime contributions to software development, management, and research. Wiley, New York (Reprinted edition)
25. Palvia P, Patula A, Nosek J (1995) Problems and issues in application software maintenance management. J Inf Technol Manag VI(3):17–28
26. Kroll P, Cantor M (2009) Software and system development with the IBM measured capability improvement framework. IBM white paper
27. OpenUP Development Framework (2007) http://epf.eclipse.org/. Accessed 27 Jan 2010
28. Kroll P, Kruchten P (2003) The rational unified process: made easy. Addison-Wesley, New York
29. Williams LA, Kessler R (2000) All i really need to know about pair programming i learned in kindergarted. CACM 43(5):109–114
30. Constantine LL (1995) Constantine on peopleware. Yourdon Press, Englwood Cliffs
31. Buchalcevova A (2009) Research of the use of Agile methodologies in the Czech Republic. In: Information systems development challenges in practice, theory, and education. Springer, New York, pp 51–64. doi: 1007/978-0-387-68772-8_5
32. Prochazka J (2008) Experience from Agile adoption in distributed environment. In: Software development conference. VSB-TU Ostrava, pp 156–163
33. Poole CJ, Huisman JW (2001) Using extreme programming in a maintenance environment. IEEE Software 18(6):42–50
34. Dettmer HW (1997) Goldratt's theory of constraints. ASQ Quality Press, Milwaukee
35. Eclipse Way (2003). http://www.eclipse.org. Accessed 21 Jan 2010
36. Doran G (1981) There's a S.M.A.R.T. way to write managements' goals and objectives. Manag Rev 70(11):35–36 (November)
37. Park RE (1996) Goal-driven software measurement. CMU/SEI-96-HB-002

Towards a Framework for Building Theory from ISD Practices

Tero Päivärinta, Kari Smolander and Even Åby Larsen

Abstract The paper presents a framework for building theory from ISD practices. The framework locates ISD practices in a learning loop that is situated in a development context. The framework recognizes that ISD practices are related to their learned rationale that may come from previous experiences, i.e. observed impacts of practices, or from existing theory. These concepts recognized by the framework are needed for building theory from ISD practices, for designing research approaches for studying ISD, for evaluating existing research on ISD practices and for evaluating ISD methods. The framework is also used in the analysis of three recent studies on ISD practices and a discussion about the uses of the framework in research and in method development is included along with a set of possible research paths in the future.

1 Introduction

Since the 1970s numerous systems development methods (SDMs), supposed to guide the practice of information systems development (ISD), have been constructed (Jayaratna 1994; Avison and Fitzgerald 2003). Traditionally, SDMs have been regarded by default as useful for the systems development process (Fitzgerald 1996). Having its roots in the late 1970s, the "reflective systems development" approach (Mathiassen 1998) was one of the first research programs which challenged the belief that development methods as such would improve systems development practice (Iivari and Lyytinen 1998). It focused on improving actual development practices in development organizations. The approach was developed through action research pursuing local improvements based on contextual circumstances (Mathiassen 1998). Since, an increasing number of systems development researchers have argued that SDMs are largely adapted during their adoption and use (Stolterman 1992; Fitzgerald

E. Å. Larsen (✉)
University of Agder, Kristiansand, Norway
e-mail: even.larsen@uia.no

J. Pokorny et al. (eds.), *Information Systems Development,*
DOI 10.1007/978-1-4419-9790-6_49,

1998a, b; Kautz 2004; Päivärinta et al. 2010), while there exist varying, even contradictory, rationales and pressures both for and against method use (Fitzgerald 1998b; Smolander et al. 2010).

For a reflective systems developer, a pre-defined (textbook) methodology represents mainly a framework of learning, instead of a prescribed guideline to be followed in practice (Mathiassen 1998; Madsen et al. 2006). Field research on agile methodologies seems to confirm this as well—the agile principles seem not always to fit in the actual practice without contextual adjustments, either (Fruhling and De Vreede 2006). In general, e.g. in the field of modeling methods for ISD, empirical research on method evaluation has remained to be in the minority even until recently (Siau and Rossi 2010).

However, even though the research community has taken a step away from the ideal of pursuing universal methods to guide the practice (Fitzgerald 1996; Mathiassen 1998), several problems in systems development seem to stay the same (Kautz et al. 2007). Moreover, lack of organizational learning has hindered improvements in ISD practice (Lyytinen and Robey 1999; Wastell 1999; Birk et al. 2006). Whether we agree with the statement that practice has led research (Glass 1995) or not, there is little doubt that practice has at least continuously deviated from the normative research in many areas of systems development for a long time (Mathiassen 1998). Calls for research on systems development *practices* (or, in other words, methods-in-action), instead of normative studies on methods, remain relevant (Wynekoop and Russo 1997; Fitzgerald et al. 2002; Kautz et al. 2007).

Already in 1997, Wynekoop and Russo called for practice descriptions to supplement ISD research in IS journals and conferences, and to form a database of them in order to identify emerging patterns of ISD behavior and outcomes. Gasson (1999) argues that detailed simulations of design contexts, rather than abstract methods, are needed for learning IS development and design skills. A number of ISD practice descriptions have been published in prominent IS journals during the last decade (e.g. Fitzgerald et al. 2003; Madsen et al. 2006; Fruhling and De Vreede 2006; Berger and Beynon-Davies 2009). However, they have not followed any common framework to accumulate the knowledge. This article aims at, by building upon the previous literature on ISD research and research on ISD practices, establishing a framework for building theories from ISD practice descriptions, aiming at possibilities for accumulating knowledge from them for teaching and research purposes. Indeed, reforms in IS education from plain method-centered teaching towards understanding the practice of IS development is seen as one of the strategies to overcome the learning failure that has plagued the field of ISD for decades (Lyytinen and Robey 1999).

The rest of the article is structured as follows. First, we introduce the framework that can be used for structuring research and observations on ISD practices. After that, we analyze existing research (Berger and Beynon-Davies 2009; Madsen et al. 2006) in light of our framework and show to what extent they succeed in analyzing and observing the components in the framework. Finally, we discuss the possible uses of the framework and end the paper with conclusions and a suggestion for further research.

2 Framework

Our framework builds on six main concepts which need to be distinguished in order to learn from ISD practice and to build theories of ISD practices: *learning, a practice, development context, rationale, impact,* and *theory*. The first part of this section gives a brief definition of these concepts separately, based on the previous literature. The second parts ties them together as a framework through which theories of ISD practices can be built by learning from practice.

Lyytinen and Robey (1999) relate *learning*, based on Argyris and Schön (1996), to the idea of "theories-in-use". To learn from practice requires that we identify or assume "causal relationships between actions taken during ISD and desired outcomes" (Lyytinen and Robey 1999, p. 88). Learning from a particular set of systems development actions requires that we treat systems development projects and actions as "experiments" from which we generate evidence to test selected theories-in-use with regard to selected ideas of ISD practices (ibid.).

A central concept in our framework is the concept of *a practice*. One dictionary definition of a practice is "something people do regularly" (Collins Cobuild English Dictionary 1989). In context of a development project or an organization, a systems development practice may become an *organizational practice* or *routine*, which can be defined as the organization's routine use of knowledge, especially "know-how" (Kogut and Zander 1992). The concept of "best practices" illustrates an assumption that abstractions of such know-how can be usefully analyzed and lessons learned from practice can be transferred through them between organizational contexts and over time (e.g. Szulanski 1996). However, organizational practices often have tacit components embedded partly in individual skills and partly in collaborative social arrangements (Kogut and Zander 1992; Nelson and Winter 1992; Szulanski 1996). If we compare a systems development method and a practice, a method adopted in an organization always embodies a predefined practice or a set of them, whereas a practice is not always defined at the detailed level, at least with regard to all potential elements (Tolvanen 1998) of method knowledge (Smolander et al. 2010). However, although being cautious on the concept of "best practices", we share the belief (e.g. Wynekoop and Russo 1997) that practice descriptions and definitions may appear as useful to analyze more or less recurrent development actions in context, as a basis of learning from them.

A systems development effort takes place in a *development context*, which comprises the systems development, organizational, and environmental contexts (Orlikowski 1993; Ramesh 1998). For example, Orlikowski (1993) identifies that the role of IS, ISD structure and operations, ISD policies and practices, IS/ISD staff, corporate strategies, organizational structure and culture, customers, competitors, and available technologies represent contextual categories of issues which may influence changes in ISD practices (in the case of her research, adoption of computer-aided software engineering tools).

Rossi et al. (2004) discuss the concept of method *rationale* as an important part of evolutionary method engineering to support ISD. While they regard good understanding of method rationale as necessary for continuing "modification and augmentation" of an organization's methods, we will widen the concept of rationale to be as well useful for understanding reasons for an organization's IS development practices in general (i.e. also those practices in use, which do not necessarily fulfill the characteristics of a thorough method). A rationale for a development practice thus provides justifications for the creation, use and modification of the practice (or set of practices) of interest.

Lyytinen and Robey (1999) emphasize the importance of learning from the organization's own ISD experience. This requires analysis and identification of *impacts* of the practices to the system, project, or to the development context in general. Such impacts may be desired already according to the explicit method rationale(s), or they may be unexpected, sometimes even unwanted.

Finally, these concepts are needed for creating and evaluating *theories* of ISD practices. That is, we pursue theories which can analyze, describe, and explain contextual ISD practices, ultimately aiming at a level of prediction (cf. Gregory 2006). That is, we believe that it is useful to analyze the practice and aim at predictive theories of certain types of ISD practices, with regard to their impacts on the development products, projects and processes, and contexts.

In the following, let us relate these concepts to each other to form a framework to guide research on ISD practices. Figure 1 relates these concepts together whereas their relationships we need to understand in order to build theory from practice descriptions are discussed further below.

Learning (L) is a boundary-spanning mechanism which need to exist, on the one hand, in a *development context* (DC) so that previous *theories* (T, including previous, more or less well-grounded, methodological recommendations) of ISD can inform local *rationale* (R) for new practices ($T \rightarrow L \rightarrow R$) and that observed *impacts* (I) of the target organization's previous *practices* (P) can inform further local rationality to adjust the practices ($I \rightarrow L \rightarrow R$). On the other hand, learning is needed between development organizations and the theory builders, who observe development actions (and local interpretations of such actions) in practice and try to abstract lessons to be learned from the particular practices in question ($L \rightarrow T$) (Fig. 1).

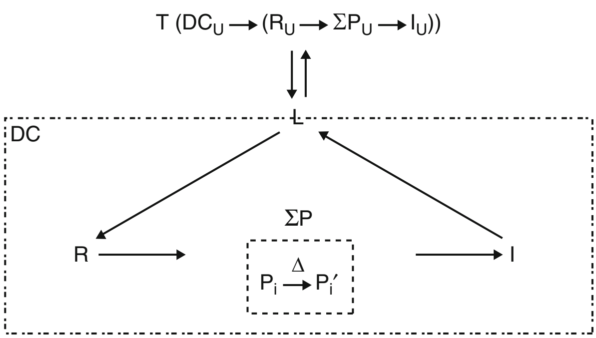


Fig. 1 A framework for building theories from ISD practices

Development context (DC) involves all the issues which have impact on how practices in the target organization or project are socially constructed and how the systems development organization can learn from its practices. The context has impact on rationale (R) to implement new practices and to motivate change, on the actual construction of practices (P) themselves, on the impacts (I) reached from the desired change, and on the learning process and lessons learned. That is, practices, their impacts, and learning may not be purely based on the identified rationale alone, but can be affected by contextual issues (Fig. 1). If contextual issues are explicitly identified before implementing a new set of practices, it becomes a part of the rationale. However, some contextual issues may have a more implicit effect on enacted practices and their impacts, recognized only after new practices have been tried out.

Learning from local and contextual development practices requires good understanding of how practices are implemented and used in any target context of development. The contextual rationale(s) for particular practices and their improvements should cause meaningful change(s) in a practice or a set of practices, which are, again, often a part of a larger, interrelated set of practices ($R \rightarrow \Sigma P$, $\Delta(P_i \rightarrow P_i')$) in the context. Moreover, contextual impact(s) after a practice has been introduced or changed need to be studied ($\Delta(P_i \rightarrow P_i') \rightarrow I$), and lessons learned from the observed impacts need to be distilled ($I \rightarrow L$) (Fig. 1).

If observed changes and improvements in local practices are used to contribute to a theory (**T**) of a selected set of general-level ISD practices (beyond the context in question) through a learning process ($L \rightarrow T$), then we need also to recognize ideas of more generic or universal rationales giving reasons to implement certain types of practices, the very ideas and descriptions of those practices and their interrelation, and ideas of impacts realized from adhering to particular practices ($R_U \rightarrow \Sigma P_U \rightarrow I_U$). As well, generic ideas to categorize development contexts, which may have impact on rationales, enactment of particular practices, and impacts resulting from particular practices, may be theorized. Through learning from the target context(s), ISD research may theorize further on more universal issues of the development context (DC_U), their impact on rationales for practices, actual practice domains of interest, and the generalized ideas of impacts from choosing particular practices ($DC_U \rightarrow (R_U \rightarrow \Sigma P_U \rightarrow I_U)$) (Fig. 1). Here, it is important to denote that the descriptions of development contexts, rationales, practices, and their impacts at the level of a theory should be distinguished from the observed practices (or local interpretations of practices) in the context. This distinction has not been always very clear in the traditional studies of methods and their use—as denoted e.g. by Fitzgerald et al. (2002).

That is, we believe that theories of ISD practices should pursue to promote understanding of reasons why to consider implementation of particular idealized practices and impact of those practices, discussed in the light of theoretical categories of contextual issues and contingencies. Such theories (T) would be able to answer to three research questions, which we believe to be of interest for IS scholars, educators, and practitioners (Fig. 1):

- Why are particular practices followed (or not) in systems development? ($R_U \rightarrow \Sigma P_U$)
- What are the expected impacts (both desired and undesired) from adhering to a set of certain pre-described practices? ($\Sigma P_U \rightarrow I_U$)
- How are certain types of development contexts expected to affect on the rationale for, the impact on, and the implementation of certain pre-described practices? ($DC_U \rightarrow (R_U \rightarrow \Sigma P_U \rightarrow I_U)$)

3 Analysis of Recent ISD Practice Descriptions in Light of the Framework

In order to illustrate the analytical usefulness of our framework, we will analyze recent and relatively detailed practice descriptions in light of it. For this purpose, we chose two relatively recent practice descriptions reported in highly profiled information systems journals (Berger and Beynon-Davies 2009; Madsen et al. 2006). Our reason for analyzing published practice descriptions is twofold. Firstly, publication in well-renowned journals implies, to an extent, research relevance of the topic, thus also implying research relevance of our framework, taken that it can be used to integrate and constructively critique such works. Secondly, published case descriptions allow anyone to replicate our analysis. However, here our purpose of analysis is first and foremost to illustrate how our framework can be used to analyze practice descriptions rather than to criticize the other research implications of the selected works under study.

An analysis in light of our framework consists of the following elements:

- $T \rightarrow L \rightarrow (R \rightarrow \Sigma P)$: What theories or methodologies and their assumptions concerning the development context and predictable impacts are reported as a background for the selected practices and as a rationale to adopt them?
- $DC \rightarrow (R \rightarrow \Sigma P,\ \Delta(P_i \rightarrow P_i'))$: What contextual issues are reported to have impact on selecting certain practices or deviating from certain initially considered practices?
- $\Sigma P,\ \Delta(P_i \rightarrow P_i') \rightarrow I$: What have been the realized impacts from the realized practices?
- $I \rightarrow L \rightarrow (R \rightarrow \Delta(P_i \rightarrow P_i'))$: What has been learned locally based on the observed impact, i.e. have the local practices been updated based on the learning?
- $I \rightarrow L \rightarrow T$: What has been theorized further to improve the theory or methodology in question or to consider in other development contexts?

Madsen et al. (2006) describe how a local ISD method emerges in practice during an in-house, academic-guided, web-based project in a market research company. Further on, they propose a technical notation to describe emergence of local practices and a framework for understanding emergence of local practices from the structuralist, individualist, and interactive process perspectives.

Table 1 An analysis of Madsen et al. (2006)

$T \rightarrow L \rightarrow (R \rightarrow \Sigma P)$	Multiview/WISDM is mentioned as a framework which influenced to the "emergent method" in the reported case. However, its influence on the realized practices during development is not made explicit in the report, beyond the initial project plan "which was not enforced" and "a job satisfaction survey" (p. 232)
$DC \rightarrow (R \rightarrow \Sigma P, \Delta(P_i \rightarrow P_i'))$	Involvement of an academic in the project → Consideration of Multiview, initial project plan framing, a "job satisfaction survey" Company's existing tools for development → Continuing use of Microsoft SQLServer and Macromedia ColdFusion as tools Technically complex web-based system → Focus on database modeling → Choosing the Erwin data modeler tool Project-team members (individual) preferences, lack of practical experience of developer, who was guided by the academic supervisor → "Time-boxed prototype-driven approach", with such techniques as E/R diagramming, use cases, flow charts, think aloud tests
$\Sigma P, \Delta(P_i \rightarrow P_i') \rightarrow I$	The report focuses on "emergence of a local IS development method", few impacts on the selected or emerged practices are reported
$I \rightarrow L \rightarrow (R \rightarrow \Delta(P_i \rightarrow P_i'))$	No explicated local learning from selection or emergence of particular practices to update the local practices over time (during the project) is reported. The report plainly describes some of the emerged practices
$I \rightarrow L \rightarrow T$	The report suggests that "managers and developers should establish a clear business vision of what a project is to achieve and should organize work around this vision rather than around fixed project plans." (p. 237). This recommendation is most likely based on an experienced impact that the choice to do this was regarded as satisfactory

An analysis of Madsen et al. (2006) in light of our framework (Table 1) revealed a few suggestions for improvement, if we would consider the article as a practice description. Firstly, although the report highlights a previous methodology (Multiview/WISDM) as a background for the project, the report makes it later on clear that it had very little impact on the realized practices. Actually, the report states that it was because the academic involvement that the methodology in question was considered in the first place. Further on, the report mentions mainly pre-existing organizational and individual habits as the main reasons to implement the most realized practices. While this may be fair enough, as the focus of the article is to plainly describe local emergence of practices, the report, however, does not explicate any impacts from the selected practices or document any local learning from the realized practices. The main contribution of the report remains to be a technical notation to describe "emergence of a local ISD method" and general-level remarks on that ISD practices emerge. The only recommendation for future ISD practice concerns establishment of a clear business vision, around which actual practices should be organized, instead of adhering to pre-planned practices or activities. However, in light of our framework which underlines learning as a key concept to improve ISD practice, the lack of grounding of such emerging recommendations on any explicit local learning experience in this case could perhaps be questioned. All in all, Madsen et al. (2006)

focus mainly on understanding how the development context impacts on the realized practices, and how that can be modeled in light of the theoretical concepts they suggest.

Berger and Beynon-Davies (2009) report how principles of rapid application development (RAD) were applied in a large and complex public-sector project, despite of the common wisdom speaking against of doing so (Table 2). The RAD-like practice was adopted due to the previous experience of the development company. The analysis in light of our framework shows that the longitudinal practice description is able to explicate both local learning renewing the adopted practices and lessons to be learned at the level of the RAD practice in general. In this case, the rationale to adopt a set of RAD practices was simply due to previous experience of the development contractor and no other scrutiny whether the practice would fit to the project from start was done. However, the initially experienced mismatch between the development context and selected set of practices lead to interesting research results, also partially because of unexpected contextual changes. This practice description appeared to cover all the elements of our framework, and it does not introduce other practices than those from which learning took place during this case. However, if a practice description would follow our framework, the learning sequences may perhaps be made even more explicit in the reports, showing the rationales (and lack of rationales) for practice choices, impacts, and loops of learning, finally resulting lessons to be brought back to theory.

4 Discussion

As stated in the introduction, a large number of studies of system development practice show that formal system development methods, if used at all, are not followed slavishly, but adapted to the local context. This is a problem for empirical research on SDM use, because it is difficult to determine whether any impact found should be attributed to the method, or to the local changes. A similar problem is the complex interactions between the different practices, and between the practices and the context. Again, any effects found may be due to the method in question, or contextual factors. Even the study itself may be such a factor (Sommerville and Ransom 2005).

On the other hand, the relationship between a single practice and the context is somewhat simpler. If the study is focused on a single practice, the impact of the practice can be determined with a greater degree of certainty. It is even possible to do laboratory experiments on some practices (Arisholm et al. 2007). This adds our knowledge of the direct impacts of the practice in a normalized research context and leads to a more complete picture of how the context influences the practice and its impact. However, it is difficult to take long-term organizational learning and practice development into account in experimental research, which is explicitly presented in our framework.

Our framework highlights the need for ISD researchers to understand better how learning affects the ISD practices, both in order to improve local practices further

Table 2 Analysis of Berger and Beynon-Davies (2009)

$T \rightarrow L \rightarrow (R \rightarrow \Sigma P)$	The article highlights some principles typically related to RAD (pp. 553–554) RAD would fit to applications, which are not computationally complex and which do not need to inter-operate with existing programs RAD is regarded as most suitable for small development teams, less than 10 people RAD assumes strong stakeholder involvement and decision-making, involving managers and users tightly to the development process Effective communication during development among stakeholders → Prototyping
$DC \rightarrow (R \rightarrow \Sigma P, \Delta(P_i \rightarrow P_i'))$	Large project (2–3 years time-line, 50 developers), RAD principles were adopted despite of this A public sector project, need to send request for bids → Pre-defined requirements catalogue, Generic Process Model Outsourced development subcontractor, despite of that co-location of developers and clients Developers' had their own in-house "RAD-like" method, of which they had good experience and previous results → Choice of "iterative application development" with incremental deliveries Bureaucratic, hierarchical and "blaming" organization culture of the client → Difficulties to implement fast and authoritative decision-making during joint application development, lack of trust between developers and business managers, rather vertical than horizontal communication, non-easy diffusion of critical business decisions The European Commission (EC) reform caused major change to initial requirements during the second year of the project → Increased acceptance of the iterative principles after the first year's difficulties to adopt the "RAD" way-of-working
$\Sigma P, \Delta(P_i \rightarrow P_i') \rightarrow I$	Lack of trust between developers and managers → No common vision of development, no joint team identity or team spirit, developers focused on beforehand-delivered requirements not changing the focus from them, "us and them" –rhetoric, difficulties in specifying requirements, larger amount of development iteration and testing Inability of developers to make empowered decisions about business needs → Lack of requirements prioritization, inability to meet time-boxed deadlines Application of iterative principles → Kept the project "alive" after the requirements changed radically due to the EC reform
$I \rightarrow L \rightarrow (R \rightarrow \Delta(P_i \rightarrow P_i')$	Lack of trust with its impacts → Managers speculated retrospectively that those could have been avoided with stronger and more direct supervision Lack of requirements prioritization, inability to meet time-boxed deadlines → Senior management created the roles of business champions, empowered to make decisions of their business areas of responsibility (which was not uniformly regarded as successful later, either) The project survived "alive" due to the iterative process when the requirements changed radically → Increased acceptance of the RAD principles among business managers, gradual change in the hierarchical organization culture
$I \rightarrow L \rightarrow T$	Confirms that if the context does not fit the assumptions of RAD, several problems will arise in the short term In a large-scale project, however, there may be enough time so that the RAD practices can diffuse and infuse even within the initially non-fitting context ISDM adoption is a dynamic and continuous process that affects both the ISDM itself and the organization in which it is applied The uncertainty of requirements in long projects may outweigh initial structural problems to adopt the principles of the RAD approach

as well as to improve abstractions of certain practice collections (such as theories or certain textbook methodologies). The framework contributes by making elements to be included in such learning more explicit, which is useful for further reports describing particular practices or sets of practices in particular contexts and explicating learning from such cases.

For ISD method developers the framework has many things to offer. First, it provides an analytical framework that makes a distinction between theory (the formalized method, Fitzgerald et al. 2002) and local implementation visible. Method developers could be more explicit in ways of how the method and its defined practices can be locally implements.

Secondly, the framework makes the learning loop explicit. Method developers should create explicit mechanisms for adjusting the methods based on learning and observed impacts. In maturity models, such as CMMi and SPICE this is taken into account at higher maturity levels, but the method literature does not seem to support this fully. For method developers, as for researchers, the framework offers a possibility to analyze the impacts of the practices defined by the method. Many approach method evaluation through experimenting with single practices. These experiments add our knowledge with direct impacts that a method produces. However, method and practice development is a continuous and long term effort where the effectiveness of the learning loop is important. This kind of an approach is already in the philosophy of some agile methods. For example, some versions of extreme programming include the rule “Fix XP when it breaks” (Wells 2009), which implies a learning loop similar to our framework.

Thirdly, the framework shows explicitly that the development context has an effect on practices. Perhaps method developers should concentrate more on giving practitioners a set of quite generic guidelines and practice templates supported with a more detailed set of instruction examples in various contexts and a tested procedure for observing the impacts and changing the practices according to lessons learned.

5 Conclusions and Further Research

In this paper we have presented a framework that locates ISD practices in a learning loop that is situated in a development context. The framework recognizes important concepts that are needed for building theories, designing research approaches and evaluating existing research on ISD practices. We also showed the use of the framework in the analysis of two recent studies on ISD practices and discussed the uses of the framework in research and in method development.

The framework creates possibilities for novel research approaches in ISD. They are related both to observing the practice of ISD and to observing the research, theory and method development in ISD. An important path of research is to increase our understanding of ISD practice and observe explicitly the learning loop in living ISD organizations using the concepts from the framework. Another path of research would be to combine the framework with tools to recognize the enactment of individual practices (such as the NIPO grid, Smolander et al. 2010) and come to more fine-grained conclusions of practice use and impacts. Yet another research opportunity is to observe and classify existing research and ISD methods in light of the framework and come to conclusions about which parts are really studied and what are their combined conclusions. One possible result of such effort could be a database of practice descriptions and patterns in ISD (Wynekoop and Russo 1997), which would make the theory component explicit and well organized according to the structure defined by our framework.

References

Argyris C, Schön DA (1996) Organizational learning II. Addison-Wesley, Reading

Arisholm E, Gallis H, Dybå T, Sjøberg DIK (2007) Evaluating pair programming with respect to system complexity and programmer expertise. IEEE Trans Soft Eng 33(2):65–86

Avison DE, Fitzgerald G (2003) Information systems development: methodologies, techniques and tools, 3rd edn. McGraw-Hill, New York

Berger H, Beynon-Davies P (2009) The utility of rapid application development in large-scale, complex projects. Inf Syst J 19:549–570

Birk A, Dingsøyr T, Lindstaedt SN, Schneider K (2006) Learning software organizations and requirements engineering: first international workshop. J Univers Knowl Manag 1(2):59–68

Collins CoBUILD (1989) English dictionary

Fitzgerald B (1996) Formalized systems development methodologies: a critical perspective. Inf Syst J 6(1):3–23

Fitzgerald B (1998a) An empirical investigation into the adoption of systems development methodologies. Inf Manag 34(6):317–328

Fitzgerald B (1998b) An empirically-grounded framework for the information systems development process. In Proceedings of the International Conference on Information Systems (ICIS). AIS, pp 103–114

Fitzgerald B, Russo NL, Stolterman E (2002) Information systems development: methods in action. McGraw-Hill, London

Fitzgerald B, Russo NL, O'Kane T (2003) Software development method tailoring at Motorola. Commun ACM 46(4):64–70

Fruhling A, De Vreede G-J (2006) Field experiences with eXtreme programming: developing an emergency response system. J Manag Inf Syst 22(4):39–68

Gasson S (1999) A social action model of situated information systems design. DATA BASE Adv Inf Syst 30(2):82–97

Glass RL (1995) Software creativity. Prentice-Hall, Englewood Cliffs

Gregory S (2006) The nature of theory in information systems. MIS Q 30(3):611–642

Iivari J, Lyytinen K (1998) Research on information systems development in scandinavia—unity in plurality. Scand J Inf Syst 10(1):135–186

Jayaratna N (1994) Understanding and evaluating methodologies: NIMSAD, a systematic framework. McGraw-Hill, Inc., New York

Kautz K (2004) The enactment of methodology: the case of developing a multimedia information system. Proceedings of the International Conference on Information Systems, Washington DC, USA

Kautz K, Madsen S, Nørbjerg J (2007) Persistent problems and practices in information systems development. Inf Syst J 17(3):217–239

Kogut B, Zander U (1992) Knowledge of the firm, combinative capabilities, and the replication of technology. Org Sci 3(3):383–397

Lyytinen K, Robey D (1999) Learning failure in information systems development. Inf Syst J 9:85–101

Madsen S, Kautz K, Vidgen R (2006) A framework for understanding how a unique and local IS development method emerges in practice. Eur J Inf Syst 15(2):225–238

Mathiassen L (1998) Reflective systems development. Scand J Inf Syst 10(1&2):67–118

Nelson RR, Winter SG (1982) An evolutionary theory of economic change. Belknap Press, Cambridge

Orlikowski W (1993) CASE tools as organizational change: investigating incremental and radical changes in systems development. MIS Q 17(3):309–340

Päivärinta T, Sein MK, Peltola T (2010) From ideals towards practice: paradigmatic mismatches and drifts in method deployment. Inf Syst J 20:481–516

Ramesh B (1998) Factors influencing requirements traceability practice. Commun ACM 41(12):37–44

Rossi M, Ramesh B, Lyytinen K, Tolvanen JP (2004) Managing evolutionary method engineering by method rationale. J AIS 5(9):356–391

Siau K, Rossi M (2010) Evaluation techniques for systems analysis and design modelling methods—a review and comparative analysis. Inf Syst J 21:249–268

Smolander K, Larsen EÅ, Päivärinta T (2010). Explaining change paths of systems and software development practices. In Proceedings of the 18th International Conference on Information Systems Development (ISD 2009), Nanchang, China, Sept 2009

Sommerville I, Ransom J (2005) An empirical study of industrial requirements engineering process assessment and improvement. ACM Trans Softw Eng Methodol 14(1):85–117

Stolterman E (1992) How system designers think about design and methods: some reflections based on an interview study. Scand J Inf Syst 4(1):137–150

Szulanski G (1996) Exploring internal stickiness: impediments to the transfer of best practice within the firm. Strategic Manag J 17(Winter Special Issue):27–43

Tolvanen JP (1998) Incremental method engineering with modeling tools: theoretical principles and empirical evidence. Jyvässkylä Studies in Computer Science, Economics and Statistics. University of Jyväskylä

Wastell DG (1999) Learning dysfunctions in information systems development: overcoming the social defences with transitional objects. MIS Q 23(4):581–600

Wells D (2009) Extreme programming: a gentle introduction. http://www.extremeprogramming.org. Accessed 1 April 2010

Wynekoop JL, Russo N (1997) Studying system development methodologies: an examination of research methods. Inf Syst J 7:47–65

Role of the Concept of Services in Business Process Management

Vaclav Repa

Abstract This paper deals with the role of the concept of services in the area of Business Process Modeling and Management. It describes the process of the business processes system design which is the part of the Methodology for Business Processes Analysis and Design—MMAPB. The design technique covers the whole process from identification of the basic activities over the design of key and supporting processes as late as the building resulting infrastructures. The very significant tool for the structuring of the processes is the principle of services. It allows discovering of basic supporting processes in bodies of key processes, their clarification with the exact definition of the interfaces between processes, and, finally, exact definition of the needs and possibilities of supporting infrastructures. The paper argues for the idea that thinking in terms of services is much more useful and general principle to be limited to the area of technology and software systems development only.

1 Introduction—Building the Process Managed Organization

1.1 Context

The first complete explanation of the idea of process management as a style of managing an organization has been published in [3]. The authors excellently explain the historical roots, as well as the necessity, of focusing on business processes in the management of the organization. The major reason for the process-orientation in management is the vital need for the dynamics in the organization's behavior. It has to be able to reflect all substantial changes in the technology as well as in the market as soon as possible. The only way to link the behavior of the organization to the changes in the market and technology possibilities is to manage the organization as a set of processes principally focused on customer needs. As customer needs, as well

V. Repa (✉)
Department of Information Technology, University of Economics
W. Churchill sqr. 4, 130 67, Prague 3, Czech Republic
e-mail: repa@vse.cz

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_50,

as requirements driven with the technology possibilities, are constantly changing the processes in the organization should change as well. That means that any process in the organization should be linked to the customer needs as directly as possible. Thus, the general classification of processes in the organization distinguishes mainly between:

- *Key processes*, i. e. those processes in the organization which are linked directly to the customer, covering the whole business cycle from expression of the customer need to its satisfaction with the product/service.
- *Supporting processes*, which are linked to the customer indirectly—by means of key processes which they are supporting with particular products/services.

Whilst the term "key process" typically covers the whole business cycle with the customer—it is focused on the particular business case; the supporting process is typically specialized just to the particular service/product, which is more universal—usable in a number of business cases. This approach allows the organization to focus on the customers and their needs (by means of the key processes), and to use all the traditional advantages of the specialization of activities (by means of the supporting processes) at the same time. Key processes play the crucial role—by means of these processes the whole system of mutually interconnected processes is tied together with the customers' needs. Supporting processes are organized around the key ones, so that the internal behavior, specialization, and even the effectiveness of the organizations' activities are subordinated to the customers and their needs.

1.2 Problem

Once we accept the ideas of process managed organization stated above we need to answer the set of consequential questions about how to design such a system:

- "how to find the right structure of business processes which respects professional standards to be able to fully exploit possibilities of outsourcing, and, at the same time, is driven by the naturally changing as well as naturally specific key processes?"
- "how to design the organizational aspects as the system of competencies and responsibilities, working positions, organization structure, etc. which are explicit, understandable, and are working as a standard, and, at the same time, which are flexible enough to be able to fully support naturally changing processes?"
- "how to design the information system which maximally follows the technology standards in order to fully exploit its possibilities, and, at the same time, which is flexible enough to be able to support naturally changing processes in full detail?"

The view of the behavior of a process-managed organization is quite different from the traditional one. Mainly, the key processes represent an unusual view of communication and collaboration within the organization. In traditionally managed organizations the organization structure reflects just the specialization of work; it is static, and hierarchical. The concept of key processes brings the necessary dynamics to the system—key processes often change according to the customer needs, while supporting ones are relatively stable (the nature of the work is relatively independent of the customers' needs). At the same time, the key processes represent the most specific part of the organizations' behavior, while the supporting ones are more general and standard. Thus, the supporting processes are the best candidates for possible outsourcing while the key ones should be regarded, rather, as an essence of the market value of the organization. So, we have a system of processes with very different nature. To ensure the necessary communication among them, we need to have the interface which enable overcoming of these differences.

Similar problem we can see also with the interface among the system of business processes and its supporting infrastructures: organizational structure, and information system. In both cases we have to harmonize different systems working differently, with different goals, and under different circumstances (see the questions at the beginning of this sub-section).

1.3 Solution

The starting point for all infrastructures (i.e., organization structure, and information system) is always the real structure (i.e., structure of business processes in the organization). So that the right solution of the main problem—coordination and harmonization of business processes—must be the starting point for the solution of consequential problemswith infrastructures. Coordinating business processes we, at the same time, create the basis for both: harmonization of substantially different interests and responsibilities of different organizational roles, as well as harmonization of substantially different business processes versus processes of information system.

This paper proposes to use the concept of Service as a common general solution of all the problems stated in the previous sub-section. It argues for *thinking in terms of services* when analyzing the interface among business processes. Firstly, we need to define the parameters of each connection point of two processes with respect to both sides of this relation—as the *service offered by the supporting process to the supported one*. This way the idea of a process managed organization perfectly fits the idea of the service-oriented structure of a system. Moreover, such a way created system of business processes is the best basis for creating naturally harmonized infrastructures as it is shown in the following section.

The paper is organized into the four main sections. After this introduction to the problem and the root idea of "thinking in terms of services" the procedure of the business processes system design is roughly described in the Sect. 2. This procedure uses the service as a crucial point for all main decisions about the structure of the system of business processes as well as about its supporting infrastructures.

Concept of services, as a part of the Business Systems Modeling Meta-model, is elaborated in detail as a specific topic in the Sect. 3.

Final Sect. 4 contains some conclusions, information about the experimental evaluation of the methodology and about intended further work in the future.

2 How to Design the System of Business Processes

In this section we describe the process of the Business Processes System Design which is a part of the Methodology for Business Processes Analysis and Design—MMAPB. The design technique covers the whole process from the identification of the basic activities to the design of key and supporting processes as late as the building of the resulting infrastructures. The tools used by the methodology are based on common standards BPMN [1], UML [12, 13], and Eriksson/Penker Notation [2].[1] The root of the methodology is defined in the formal meta-model [10] as a part of the development project OpenSoul [9]. The key ideas of the modeling method are described in [8], and [7]. Figure 1[2] illustrates how the above mentioned standards are used in the methodology for the global and detailed view on business processes.

Global view of all processes is represented by the Global process model of an organization (see Fig. 1). Global model is always complete which means that it identifies all important processes of the organization. In fact the Global process model represents the object view of processes; it describes the existence of processes and their mutual relationships. This means that the Class Diagram from UML would be the proper diagram for this model. Nevertheless, asall processes have attributes of thesame type, all relationships are oriented and with the same meaning, and there are just small number of other complementing types of objects (goal, document, product, . . .) some specialization of the Class Diagram is more suitable. So that we use the Eriksson-Penker specialization of UML [2] for the Global Process Model.

Global view is complemented with possible detailed view of selected processes (usually all key processes, and some important supporting processes). This detailed view, compared with the global one, is partial only, it shows just one process. Nevertheless it allows to see the process details in the process manner—unlike the global view it describes the run of the process. We use the Business Process Modelling Notation [1] for this model.

The important piece of knowledge of the presented methodology is the fact that it is necessary to view the process system from both: global, as well as detailed perspectives where both perspectives are mutually incommutable. It is impossible to describe all processes in the organization including their mutual relationships using just the BPMN. It is because with the BPMN it is possible to describe just the process run, but not the full context of the process (i.e., all it's communication with other processes). So we always need to complement the process-oriented view of

---

[1] In fact, any other suitable notation, like Aris for instance, can be used in place of the BPMN and Eriksson-Penker Notation. The reason for the use of these notations is the fact that they are standardized with principal respect to the UML which is the widely respected standard used in both: analytical as well as IS Development activities. This fact we regard as very important in the MMABP methodology. Complete methodology content together with the definition of requirements for diagrammatic tools can be found in [9], and [10].

[2] This figure is just an illustration of different views and diagrams. Unreadable text is not important.

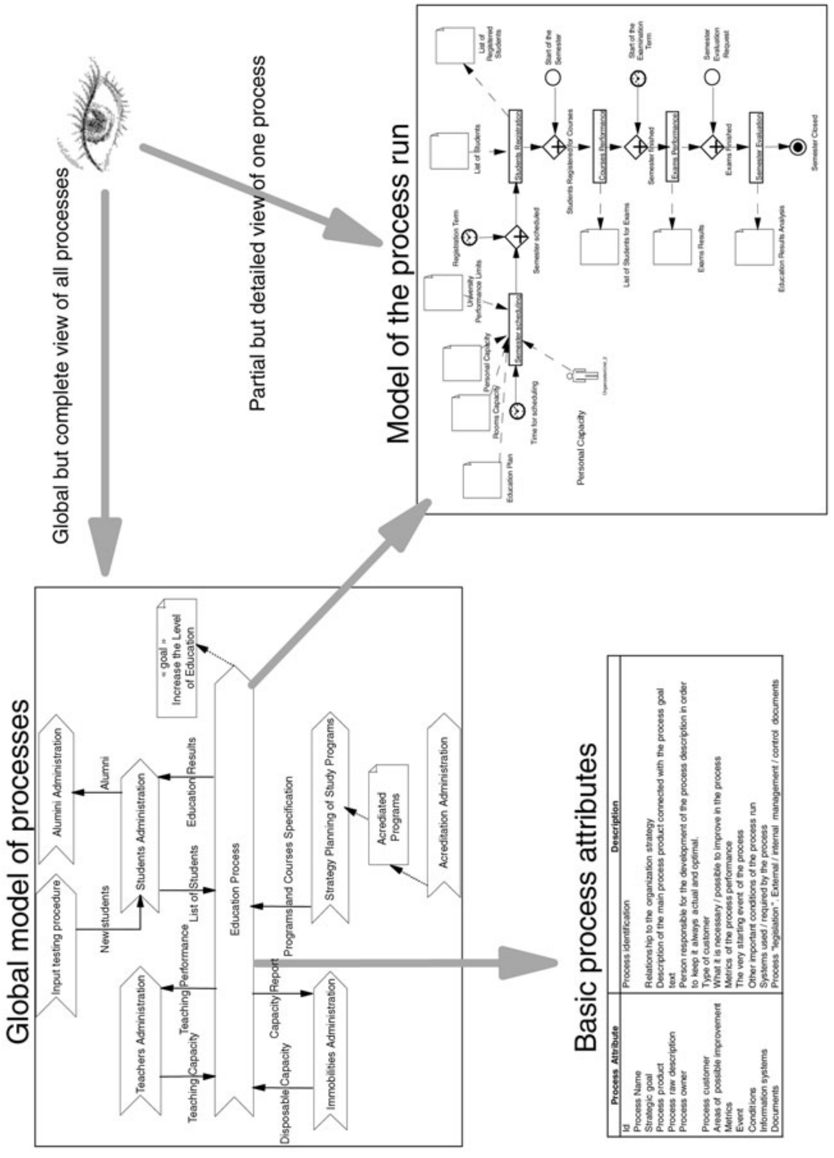


Fig. 1 Global versus detailed view on process

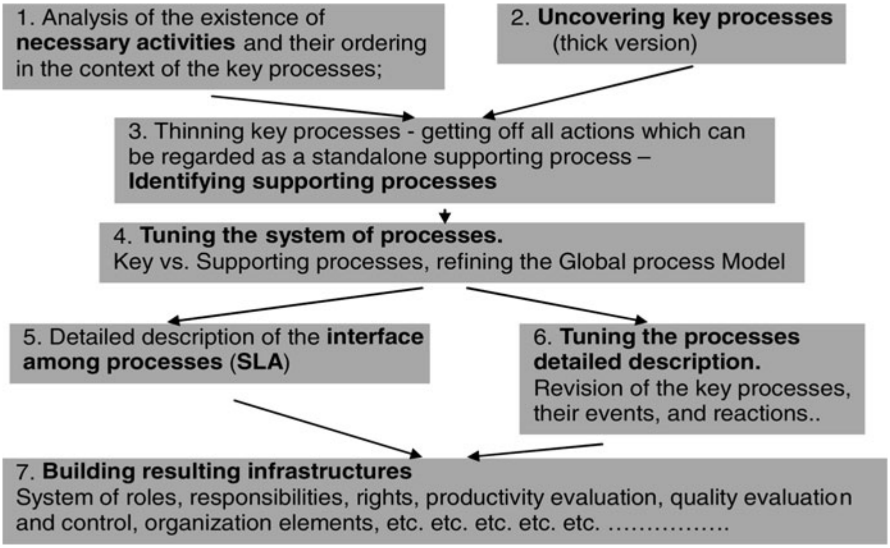


Fig. 2 Procedure of the business processes system design

the process with the object-oriented view (represented here by the Eriksson-Penker Notation) describing the global context of processes.[3]

The crucial role in the presented methodology plays the *concept of services*. The importance of this concept follows from several mutually connected phenomena: from the historical roots of Service Level Agreements in the area of management [5], over the Service Oriented Architecture in IS/IT [6], up to the latest generalized Service Science Management and Engineering (SSME) [11]. In the methodology we use thisconcept as a main tool for structuring activities in the processes as well as for building the interface between the process model (virtual world) and the real world infrastructures—organization and technology including the IS/IT.

Figure 2 expresses the procedure of the business processes system design as a set of succeeding/parallel steps:

Firstly, the basic natural sequences of activities are revealed (Step 1) together with uncovering the key processes at the same time (Step 2).

The main subject of interest in the *Step 1* is the natural succession of activities in the regular form as people know them, work flows, legal procedures etc. These sequences of activities serve in later steps as the basis for revealing the proper structure of the process system. Necessary activities (and their basic causal consequences) form the basis of the supporting processes. At the same time, they are the roots of the key processes as well (see the *Step 2*). Each key process represents the way of achieving the key type of product. The structure of the key process can, thus, be derived from

[3] In fact, there is the only way how to include all processes into one model using BPMN: to subordinate all the processes to the one of them as sub-processes. This idea however absolutely contradicts with the main idea of the Process Management, and thus it is not relevant anyway.

the life cycle of the key product, as a final result of the process. The key process is a process by which the organization realizes some external value represented by this key product—value for its customers.

For example, in the world of education it is obvious that the key product of the university is the education (which exactly means the positive increment of the knowledge/ability to know). That means that the key process of the university is the education process, obviously. On the other hand, the key product of the university is obviously not the Study program accreditation as it does not realize any external value in itself—it rather helps the key process; 'education'; to realize the value of education.

In the first version of the model, the key processes naturally contain a number of supporting activities and sub-processes. It is the important task for following steps to free key processes of all supporting activities (see below).

The outputs of the first two steps are:

- Structured list of potential processes/activities and their successions distinguishing mainly between the key and supporting processes;
- Basic attributes of the key processes;
- First version of the key process run models (basic process logic of the activities still containing a number of supporting activities which should be removed later).

Once, in the first two steps, the basic process structure is uncovered it needs to be restructured in order to find the natural border between the leading—key activities which represent the management, and the supporting ones, representing the production. Every key process is naturally "long" because it covers the whole business case from the identification of the customer need till the satisfaction of this need by the product (service). At the first sight (Step 2) the key processes are also "thick"—they contain a number of supporting activity chains. The *Step 3* strives to remove, as much as possible, the supporting activities from the key processes. We are speaking about the "thinning" of the key processes.

All action chains which can be regardedas supportingchains, arise from the key processeson the principle of "outsourcing" them into standalone supporting ones (even, possibly, outside the organization): Any relatively standalone, continuous, homogeneous, and generalizable part of the process will be removed from the key process, generalized, and established as a supporting process. As a result of this removal some control activity (managing the supporting service delivery) remains in the key process in the original place of the removed supporting activity chain.

In the step 5 (Detailed description of the interface among processes), the interface to the original (mother) key process will be described including the basic parameters of the product/service (see the principle of SLA below).

Subsequently it is necessary to adequately rework and elaborate models in detail, and complete the structure of the Global process model in the *Step 4*.

The outputs of the second two steps are:

- the second version of the key process run models expressing the key process basic logic without the supporting activities.

- Set of supporting processes newly discovered by removing supporting activity chains from the key processes.
- Completed Global process model;
- Completed specification of the interface among the processes (events, end states and their successions, and other important information for the SLA specification in the following step).

Interfaces among processes, which have arisen in steps 3–4 must be elaborated in detail. Great attention should be paid to the interface among key and supporting processes. In the *Step 5* every interface is described in the form of "SLA" (Service Level Agreement—see the Sect. 3 Concept of Services). Simultaneously the revision of the key processes run, their events, reactions, and tuning of this description with the Global model, should be performed in the *Step 6*. Models are completed with actors, inputs, outputs; and overall revision of the Global model of processes is made in this step. The steps 5 and 6 result in following outputs:

- SLA description of most important interface among processes.
- Actualized Global model of processes.
- Actualized process attributes descriptions.
- Actualized process run descriptions of the key processes and other important processes.

The final *Step 7* of the procedure represents the interface to the subsequent activities of the organization building process. This step consists of the elaboration of the process interface in order to analyze the possibilities for realizing the service which the supporting process represents (the supplier part of the SLA). Activities of this step lead to the creation of both main infrastructures:

- the basic requirements for the organizational infrastructure are analyzed with the definition of roles, their responsibilities, communication procedures, and other organizational aspects which follow on from the mutual competencies of both attendees of the business relation represented by the SLA;
- Similarly, the technical infrastructure needs can be specified this way (necessary production and workflow technology support, as well as the necessary Information System services).

Particular outputs of this step are following from the detailed specification of products/services in the SLAs, for instance: system of evaluation of the process' cost (cost-based price of the process product); system of evaluation of the process' performance; system of evaluation of the actors' performance; outsourcing decision support system; information requirements/needs of the processes, etc.

3 Concept of Services

Conceptual model at the Fig. 3 shows the fragment of the universe of services in the context of business processes. It is necessary to distinguish between the *Business*

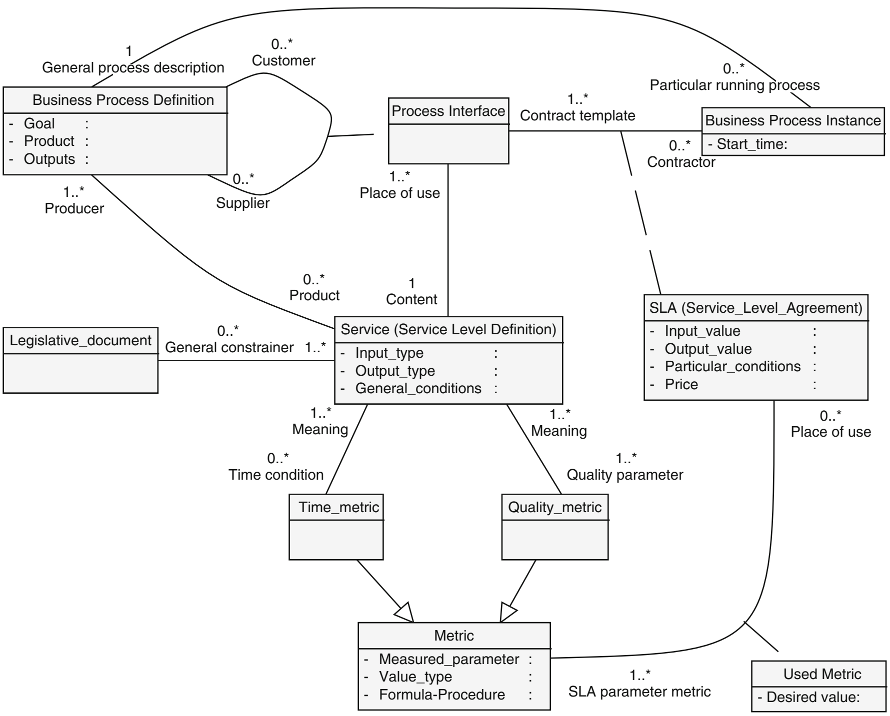


Fig. 3 Process services conceptual model

Process Definition, and the *Business Process Instance* which represents the particular business process running in particular time with particular actors, inputs, outputs, etc. Each business process can collaborate with another process. General view of the collaboration of processes is represented in this model with the Process Interface class as an association between two processes where the particular process has either the role of Supplier (producer of the *Service*) or Customer (consumer of the Service). In particular contact of particular processes the *Process Interface* has the role of the Contract template.

In general, the *Business Process* can be a Producer of one or more *Services* (see the direct association between *Business Process Definition* and *Service*).

It is obvious that there must be the integrity constraint in the model which expresses the fact that there must exist the direct association between the *Business Process Definition* and that *Service* which takes the place in the *Process Interface* in which this business process exists in the role of Supplier. In other words each business process must be the producer of all services which it supplies in various contracts.

The *Service* itself may be generally constrained by even more *Legislative documents*, it must have the association to at least one *Quality metric*, and it may be associated to one or more *Time metrics*. That means that there always must be some quantitative (measurable) notion of the quality of the service, and that some services also need to be time limited (what is the specific kind of the quality, in fact). Each service metric is characterized by the measured parameter, its value type, and the method of its gathering—formula or procedure.

Every SLA should contain product description (service characteristics, its meaning, value, sense), basic product parameters in measurable units, product quality metrics (how to measure quality as a general product attribute), and the product "price" which reflects the necessary costs of the supporting process or offered service.

The Service Level Agreement concretizes the general attributes of the *Service* with the particular attributes of the *SLA*; general Input is represented in the SLA with the Input data, Output with the Output data, General conditions of the service with the Particular conditions in the contract. SLA also contains the desired values of the metrics[4] of the service and the additional attribute—price of the service—which does not exist in the general view of the service. The price of the service always must be the result of the negotiation during the agreement specification and can never be stated as a general attribute of the service as it depends on many situational factors generally called the Market. By the way—this fact also well illustrates the crucial role of the outsourcing in the process—oriented management. In fact, such a description of the interface between two processes represents the real business agreement of these processes. It means that there is no difference between the "internal" and the real outsourcing (see the Step 3 of the Procedure of the business processes system design above). Thus, this way of thinking perfectly prepares the situation for the possible outsourcing of all supporting processes where it is suitable.

4 Conclusion and Further Work

The methodology for business processes system design, presented in this paper, which is based on the "principle of services", has been evaluated in several projects during the last four years. Every use of the methodology in the project brought some new experience and positively influenced its content. That way we lastly uncovered the need for elaborating the concept of service in general, for instance. Projects covered fields of university education, production, finance, and public administration. The main knowledge from this heterogeneous experience is:

- the idea of Process Management is valid in general, for all types of organizations, thus it should not be reduced just to the area of market-oriented organizations,
- thinking in terms of services is relevant for all types of different systems harmonization: coordination of processes as well as harmonization of competencies following from the organization structure with specific process competencies needed, harmonization of the organization specific needs with general technology possibilities, etc.

[4] See the association between the SLA and the Metric. It is obvious that even there (like in the case of relationships between Business Process Definition and Service) must be the integrity constraint which expresses the fact that the metric values used in the SLA belong to the metrics generally associated to the Service.

- the traditional gap between the business conception and the applications development can be overcome by regarding so-called “user requirements” as services of information system required by the nature of processes. This view on the problem also significantly changes the traditional approach to the distribution of tasks among typical roles in the information systems development, and, at the same time, explains the “role of the application user” often discussed nowadays in the communities of applications developers.

This paper discusses the concept of services as a tool for Business Process Management. The paper just outlines some basic contingencies which follow on from the inspiration by the theory of services in the area of process management. It points out the significant similarities among different areas of possible application of the “service-oriented thinking”, such as software development, process management, outsourcing, etc. It also points out the obvious convergence of all these phenomena—outsourcing as an original area of the SLAs is the principal way of recognizing the substantial differences between the key and supporting processes which, at the same time, directly corresponds to the need to tie the system of business processes in with the enterprise strategy on one hand, and with the supporting technology on the other hand.

The basic conclusion from the previous paragraph is: the concept of services should be regarded as a general principle for recognizing the interface between two substantially different areas connected with some common sense. This paper shows how this concept works as a guide for specifying the interface between the various types of processes (key versus supporting ones) which differ mainly in the reasons and “speed”, and are mutually asynchronous. It also shows how this principle can be used for specifying the interface between the system of processes and supporting infrastructures (technology as well as organization). A similar area of application of this principle is the interface between the strategic activities and process management of the organization which is not presented in this paper. This interface is the main subject of the work of R. Kaplan and D. Norton [4]. Their theory could be also significantly extended this way.

The paper also describes basic procedure for analyzing and designing the system of business processes in the organization with respect to the consequential activities (Building the resulting infrastructures). In this way, it outlines what should be an area of future development of the methodology.

Acknowledgements The work presented in this paper is supported by Czech project GAČR 402/08/0529 Business Process Modeling.

References

1. Business Process Modeling Notation Specification (2006) Business Process Modeling Notation Specification, OMG final adopted specification, Feb 2006, dtc/06-02-01. http://www.bpmn.org/Documents/OMG Final Adopted BPMN 1-0 Spec 06-02-01.pdf
2. Eriksson HE, Penker M (2000) Business modeling with UML: business patterns at work. Wiley, New York. ISBN 978-0-471-29551-8

3. Hammer M, Champy J (1993) Reengineering the corporation: a manifesto for business revolution. Nicholas Brealey Publishing, London
4. Kaplan RS, Norton DP (2004) Strategy maps: converting intangible assets into tangible outcomes. Harvard Business School Press, Bostan
5. Rayport JF, Sviokla JJ (1995) Exploiting the virtual value chain. Harv Bus Rev 1995(Nov 01):75–85
6. Service Oriented Architecture (2006) Reference Model for Service Oriented Architecture 1.0, OASIS Standard, 12 Oct 2006. http://docs.oasis-open.org/soa-rm/v1.0/
7. Řepa V (2007) Modeling objects dynamics in conceptual models. Budapest 31.08.2007–02.09.2007. In: Advances in information systems development. Springer, New York, pp. 139–152. ISBN 978-0-387-70801-0
8. Řepa V (2008) Process dimension of concepts. In: Jaakkola H, Kiyoki Y, Tokuda T (eds) Information modelling and knowledge bases XIX. IOS Press, Amsterdam, pp. 322–329. ISBN 978-1-58603-812-0. ISSN 0922-6389
9. Řepa V OpenSoul project. http://opensoul.panrepa.org
10. Řepa V Business system modeling specification. http://opensoul.panrepa.org/metamodel.html
11. Service Science Management and Engineering. http://www.ibm.com/developerworks/spaces/ssme
12. UML (2003) UML OMG Unified Modeling Language Specification v. 1.5. document ad/03-03-01, Object Management Group, March 2003
13. UML (2004) UML Superstructure Specification v2.0 document 05-07-04, Object Management Group, 2004

A Metamodel for Modelling of Component-Based Systems with Mobile Architecture

Marek Rychlý

Abstract Current information systems tend to be distributed into networks of quite autonomous, but cooperative, components communicating asynchronously via messages of appropriate formats. Loose binding between those components allows to establish and destroy their interconnections dynamically at runtime, on demand, and according to various aspects; to clone the components and to move them into different contexts; to create, destroy and update the components dynamically at runtime; etc. Modelling of the dynamics and mobility of components brings many issues that cannot be addressed by means of conventional architecture description languages. In this paper, a metamodel for modelling of component-based systems with mobile architecture is proposed.

1 Introduction

Globalisation of information society and its progression create needs for extensive and reliable information technology solutions. Several new requirements on information systems have emerged and significantly affected software architectures of these systems. The current information systems cannot be realised as monoliths, but tend to be distributed into networks of quite autonomous, but cooperative, components communicating asynchronously via messages of appropriate formats. Loose binding between those components allows to establish and destroy their interconnections dynamically at runtime, on demand, and according to various aspects (e.g. quality and cost of services provided or required by the components); to clone the components and to move them into different contexts; to create, destroy and update the components dynamically at runtime; etc.

Modelling of the dynamics and mobility of components (i.e. the systems' dynamic or mobile architecture [11], respectively) brings issues that cannot be addressed by conventional architecture description languages. Models have to provide description

M. Rychlý (✉)
Department of Information Systems, Faculty of Information Technology,
Brno University of Technology, Božetěchova 2, 612 66, Brno, Czech Republic
e-mail: rychly@fit.vutbr.cz

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_51,

of hierarchical composition of components, description of binding of neighbouring components' interfaces as well as interfaces received via mobility, description of interfaces providing and requiring a component's functional operations, and description of interfaces with operations controlling the component's life-cycle, binding of its interfaces and its mobility.

In this paper, a metamodel for modelling of component-based systems with mobile architecture is proposed. Section 2 outlines the current state-of-theart in modelling of dynamic and mobile architectures. Section 3 describes the proposed metamodel. An application of the metamodel is demonstrated in Sect. 4 as a model of a component-based system implementing a service of a railway interlocking control system. To conclude, in Sect. 5, we summarise the contribution of this paper and outline the future work.

2 State of the Art and Motivation

There are several architecture description languages that support modelling of component-based systems (e.g. component diagrams in UML [1], language ACME [8], etc.). They allow to describe a logical (structural) view of a component-based system, i.e. basic entities, their relations and features. However, the most of them do not support mobile or even dynamic architecture.

The *component model Fractal* [5], which is a general component composition framework, provides a notation for description of dynamic architecture of hierarchically nested components [3], but without component mobility. The *component model SOFA 2.0* [6] introduces a MOF-based metamodel for dynamic architecture, which is able to describe passing references of "utility" interfaces, i.e. limited mobility of the specific type of interfaces. The *architecture description language ArchWare* [2] provides constructs to describe dynamic software architectures by means of a specific UML 2 profile [12] and their behaviour by means of π-calculus [11]. Although the ArchWare is more advanced than the previously mentioned component models, it also does not directly address component mobility.

The motivation of the approach proposed in this paper is to address component mobility and design a MOF-based metamodel for component-based systems with mobile architecture. The metamodel will support subsequent description of the systems' behaviour by means of π-calculus.

3 Metamodel

The component model for mobile architectures is described as a metamodel in the context of a four-layer modelling architecture. The metamodel is implemented in OMG's *Meta Object Facility* (MOF, [9]), which is used as a meta-metamodel. The modelling architecture comprises the four layers:

- M0: An information layer, which is comprised of the actual data objects. This layer contains particular instances of component-based systems, their runtime configurations, specific deployments of their components and connectors, etc.
- M1: A model layer, which contains models of the M0 data. The models include structure and behaviour models that describe different perspectives of component-based systems such as, for example, UML component models or communication diagrams.
- M2: A metamodel layer provides a language for building M1 models. Component models fall in this layer, as well as models of the UML language.
- M3: A meta-metamodel layer, which is used to define modelling languages. It holds a model of the information from M2, e.g. MOF.

In the context of component-based development, a specific component-based system (in layer M0) contains instances of elements from its model (from layer M1). The model contains instances from a specific component model (a metamodel in layer M2), which is described by a given meta-metamodel (in layer M3).

The proposed metamodel is defined by means of *Essential MOF* (EMOF), which is a part of MOF in layer M3. The EMOF contains packages `Basic`, `Reflection`, `Identifiers`, and `Extension`, which form a minimal set of modelling elements to define simple metamodels. *Complete MOF* (CMOF), which is second and the last part of MOF, extends EMOF by `Constructs` package from UML 2 Core [10].

The component model, as a model of layer M2, can be described by means of UML 2 diagrams in two contexts:

1. as an object diagram of instances of EMOF classes from layer M3 (entities in layer M2 are instances of classes in M3), i.e. it is described as "a model",
2. as a class diagram from layer M1 (entities in layer M1 are instances of classes in layer M2), i.e. it is described as "a metamodel".

For better clearness, the component model will be described as an UML 2 class diagram from layer M1. To reuse well-established concepts of MOF, the component model's metamodel extends EMOF classes `EMOF::NamedElement`, `EMOF::TypedElement`, and `EMOF::Operation`, which are outlined in Fig. 1. A complete and detailed definition of the EMOF classes can be found in [9].

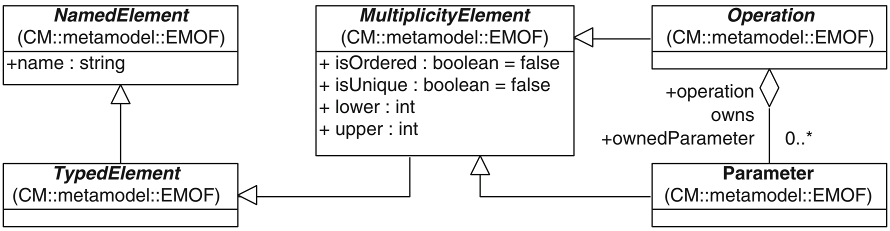


Fig. 1 A simplified part of the EMOF metamodel [9] with classes that will be extended by the component model

3.1 Components and Interfaces

Figure 2 describes the first part of the component model as an extension of EMOF. The metamodel defines an abstract component, its realisations as a primitive component and a composite component, and their interfaces. All classes of the metamodel inherits (directly or indirectly) from class `EMOF::NamedElement` in package `Basic` of EMOF.

In our approach, a *component*, which is an active communicating entity of a component-based software system, can be described form two sides: as an abstract component without considering its internal structure ("black-box" view) and as a component realisation in the form of a primitive component or a composite component ("grey-box" view). The *abstract component* (class `Component` in the metamodel) can communicate with neighbouring components via its interfaces (class `Interface`). The interfaces can be provided (class `ExternalProv-Interface`) or required (class `ExternalReqInterface`) by the component.

The component realisation can be primitive or composite. The *primitive component realisation* (class `PrimitiveComponent`) is implemented directly, beyond the scope of architecture description. It is a "black-box" with described observable behaviour (attribute `behaviouralDescription`). The *composite component realisation* (class `CompositeComponent`) is decomposable on a system of subcomponents at the lower level of architecture description (it is a "greybox"). Those subcomponents are represented by abstract components (class `Component` and relation "consists of"). Moreover, every composite component realisation can communicate with its subcomponents via its provided (class `InternalProvInterface`) and required (class `InternalReqInterface`) internal interfaces (relations "provides inside" and "requires inside", respectively).

The specific interfaces have to implement methods `getOwner()`, which return their owners, i.e. objects that act as the abstract components in a case of the abstract

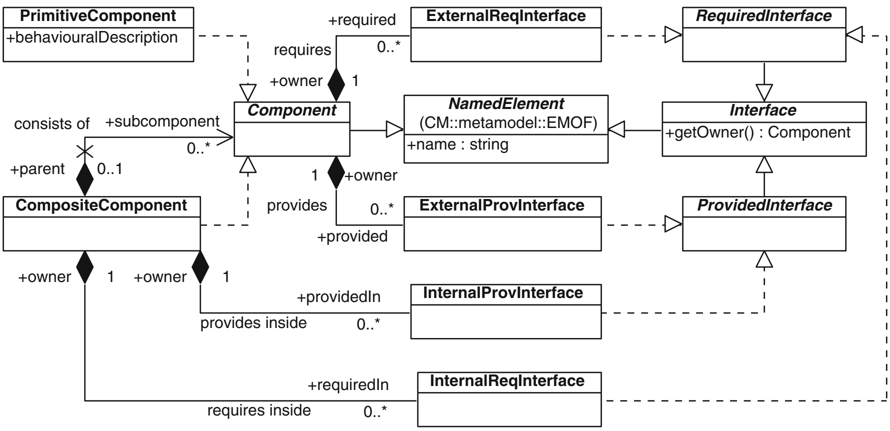


Fig. 2 Abstract component, realisations, and interfaces, extending `EMOF::NamedElement` in the metamodel of the component model

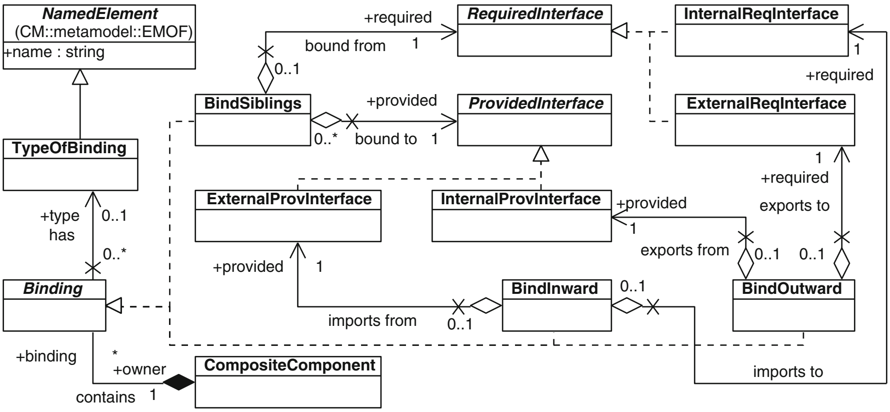


Fig. 3 Binding and its different realisations between interfaces of a composite component realisation in the metamodel of the component model. Classes `CompositeComponent` and `...Interface` are identical to the classes in Fig. 2

component interfaces or as instances of the composite component realisations in a case of their internal interfaces (in accordance with `owner` roles of components in the relations with their interfaces).

3.2 Composite Components and Binding of Interfaces

Binding is a connection of required and provided interfaces of the identical types into a reliable communication link. It is described in Fig. 3. Interfaces of a component (classes `ExternalProvInterface` and `ExternalReqInterface`) can be provided to and required from its neighbouring components, while interfaces of a composite component realisation (classes `InternalProvInterface` and `InternalReqInterface`) can be provided to and required from its subcomponents only. Therefore, we distinguish three types of the binding (the realisations of class `Binding`):

1. Binding *of provided interfaces to required interfaces in the same composite component realisation* is represented by class `BindSiblings`. The interfaces have to be internal interfaces of the composite component realisation or external interfaces of subcomponents in the same composite component realisation.[1] The binding interconnects required interfaces (class `RequiredInterface`) via relations "bound from" to provided interfaces (class `Providedlnterfaces`) via relations "bound to".
2. Binding *of external provided interfaces* of a composite component realisation *to its internal required interfaces* is represented by class `BindInward`. The external

[1] The diagram in Fig. 3 does not restrict relations of `BindSiblings` to the interfaces of the same composite component realisations; this will be defined later by means of additional constraints in Sect. 3.4.

interfaces are provided to neighbouring components of the composite component acting as an abstract component (relation "imports from" an instance of class `ExternalProvInterface`), while the internal interfaces are required from the composite component's subcomponents (relation "exports to" an instance of class `ExternalReqInterfaces`).

3. Binding *of internal provided interfaces* of a composite component realisation *to its external required interfaces* is represented by class `BindOutward`. The internal interfaces are provided to the composite component's subcomponents (relation "exports from" an instance of class `InternalProvInterface`), while the external interfaces are required from neighbouring components of the composite component acting as an abstract component (relation "exports to" an instance of class `ExternalReqInterfaces`).

The bindings (i.e. instances of the realisations of class `Binding`) are owned by the composite component realisations. Each binding can have a type (class `TypeOfBinding`), a specialisation of `EMOF::TypedElement`, which can describe a communication style (buffered and unbuffered connection), a type of synchronisation (blocking and output non-blocking), etc.

3.3 Types of the Interfaces

To ensure type compatibility of interfaces in a binding, each interfaces has a type (class `TypeOfInterface`, which is a specialisation of class `EMOF::NamedElement` in package `Basic` of EMOF). Hierarchy of the types of interfaces is described in Fig. 4.

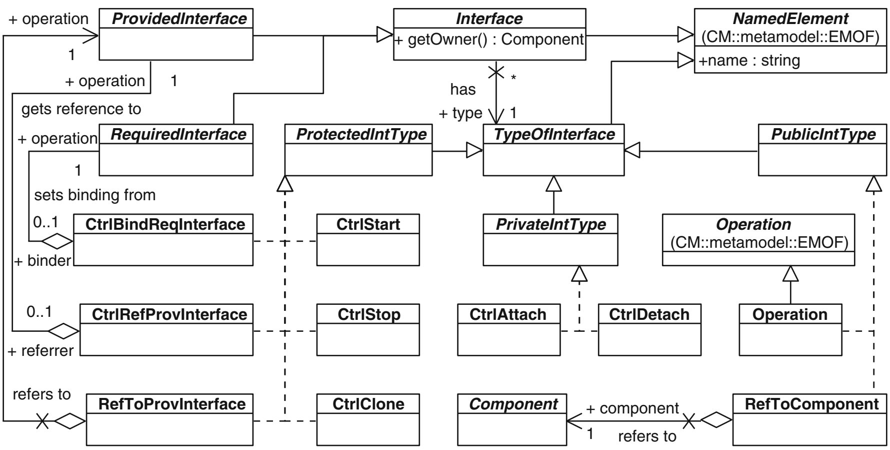


Fig. 4 Types of interfaces with class `Operation` extending `EMOF::Operation` in the metamodel of the component model. Classes `Interface`, `ProvidedInterface`, `RequiredInterface`, and `Component` are identical to the classes in Fig. 2

According to a scope of visibility of the interfaces in a composite component realisation, we can distinguish public interfaces, private interfaces, and protected interfaces. The *public interfaces* (classes realising `PublicIntType`) of a component can be accessed by its neighbouring components (via binding `BindSiblings`). If the component is a composite component realisation, its external public interfaces can be also accessed by its subcomponents and its internal public interfaces can be accessed by its neighbouring components (i.e. the interfaces can pass the component's border via binding `BindInward` and `BindOutward` owned by the component). They can be interconnected by means of all kinds of bindings.

Contrary to the public interfaces, the *private interfaces* (classes realising `PrivateIntType`) are specific types of interfaces, which can be provided only by a composite component realisation and only to its subcomponents as the component's internal interfaces.[2] They can be interconnected only by means of binding `BindSiblings`.

Finally, the *protected interfaces* (classes realising `ProtectedIntType`) of a component can be accessed by its neighbouring components as the component's external interfaces, but if the component is a composite component realisation, they are not reachable by its subcomponents. They can be interconnected only by means of binding `BindSiblings`.

We distinguish the following types of interfaces[3] by their functionality:

- Public interface `Operation`, which extends class `EMOF::Operation` from package `Basic` of EMOF and represents a business oriented service with typed input and output parameters.
- Protected interface `CtrlRefProvInterface` provides references to given provided interface `ProvidedInterface` of type `Operation`,[4] while protected interface `CtrlBindReqInterface` allows to establish a new binding of specific required interface `RequiredInterface` of type `Operation`[4] to a provided interface of another component formerly referred by `CtrlRefProvInterface`.
- Protected interfaces `CtrlStart` and `CtrlStop` allow to control behaviour of a component (i.e. to start and to stop the component, respectively).
- Private interfaces `CtrlAttach` and `CtrlDetach` provided by a composite component realisation allow to attach a new component as a subcomponent of the realisation ("nesting" of the component) and detach an old subcomponent from the realisation, respectively.
- Protected interface `CtrlClone` provides references to fresh copies of a component.
- Protected interface `RefToInterface` is able to pass references of provided interfaces `ProvidedInterface` of type `Operation`[4], while public interface

---

[2] The private interfaces can be required by the subcomponents as their external interfaces, but they cannot pass borders of the subcomponents (nor any other component). It means that the subcomponents have to be primitive components.

[3] `Operation` denotes *functional interfaces*, while the others denote *control interfaces*.

[4] The restriction to the interface of type `Operation` will be defined explicitly by additional constraints in Sect. 3.4.

RefToComponent allows to pass references of a whole component, which is required to support component mobility.

3.4 Additional Constraints

We need to define additional constraints to ensure type compatibility of interfaces in bindings, i.e. instances of realisations of class Binding in Fig. 3. Types of the interfaces are given by relation to specific instances of realisations of class TypeOfInterface and according to the hierarchy of the types of interfaces in Fig. 4. The following formulae use a first-order logic with extra predicate symbols "o : T" and "o is T" for restriction of o to type T, predicate symbol "$i \in$ L" for restriction of l to list L, predicate symbol "$x = y$" to check equality of x and y, and function symbol "i. getOwner()" to get an owner of interface i (see method getOwner() of Interface in Sect. 3.1).

1. Bindings BindInward and BindOutward in a composite component realisation can interconnect only interfaces of the same realisation.

$$
\begin{aligned}
&(\forall c : \mathrm{CompositeComponent})\ (\\
&((\forall b : \mathrm{BindInward} \in c.\mathrm{binding})(b.\mathrm{provided}.\mathrm{getOwner}() = c\ \wedge\ b.\mathrm{required}.\mathrm{getOwner}() = c))\ \wedge\\
&((\forall b : \mathrm{BindOutward} \in c.\mathrm{binding})(b.\mathrm{provided}.\mathrm{getOwner}() = c\ \wedge\ b.\mathrm{required}.\mathrm{getOwner}() = c))\ )
\end{aligned}
$$

2. Binding BindSiblings in a composite component realisation can interconnect only internal interfaces of the same composite component realisation or external interfaces of its subcomponents.

$$
\begin{aligned}
&(\forall c : \mathrm{CompositeComponent})\ (\forall b : \mathrm{BindSiblings} \in c.\mathrm{binding})\ (\\
&\quad (\forall i : \mathrm{InternalProvInt} \in b.\mathrm{provided})\ (i.\mathrm{getOwner}() = c)\\
&\quad \wedge\ (\forall i : \mathrm{InternalReqInt} \in b.\mathrm{required})\ (i.\mathrm{getOwner}() = c)\\
&\quad \wedge\ (\forall i : \mathrm{ExternalProvInt} \in b.\mathrm{provided})\ (i.\mathrm{getOwner}() \in c.\mathrm{subcomponent})\\
&\quad \wedge\ (\forall i : \mathrm{ExternalReqInt} \in b.\mathrm{required})\ (i.\mathrm{getOwner}() \in c.\mathrm{subcomponent})\ )
\end{aligned}
$$

3. Bindings Binding in a composite component realisation can interconnect only provided interfaces with required interfaces of compatible types.

$$
(\forall c : \mathrm{CompositeComponent})\ (\forall b : \mathrm{Binding} \in c.\mathrm{binding})(b.\mathrm{provided}.\mathrm{type} = b.\mathrm{required}.\mathrm{type})
$$

4. Bindings BindInward and BindOutward can interconnect only public interfaces, i.e. instances of class PublicIntType.

$$
\begin{aligned}
&(\forall c : \mathrm{CompositeComponent})\ (\\
&\quad ((\forall b : \mathrm{BindInward} \in c.\mathrm{binding})\\
&\qquad (b.\mathrm{provided}.\mathrm{type}\ \text{ is }\ \mathrm{PublicIntType}\ \wedge\ b.\mathrm{required}.\mathrm{type}\ \text{ is }\ \mathrm{PublicIntType})\ )\ \wedge\\
&\quad ((\forall b : \mathrm{BindOutward} \in c.\mathrm{binding})\\
&\qquad (b.\mathrm{provided}.\mathrm{type}\ \text{ is }\ \mathrm{PublicIntType}\ \wedge\ b.\mathrm{required}.\mathrm{type}\ \text{ is }\ \mathrm{PublicIntType})\ )\ )
\end{aligned}
$$

5. Bindings `BindSiblings` that are inside a composite component realisation can be connected to private interfaces, only if the interfaces are internal interfaces of the composite component realisation.

$$(\forall c : \text{CompositeComponent})\ (\forall b : \text{BindSiblings} \in c.\text{binding})$$
$$(b.\text{provided}.\text{type}\ \text{is}\ \text{PrivateIntType}\ \Rightarrow\ b.\text{provided} \in c.\text{providedIn})$$

6. Instances of classes `CtrlBindReqInterface`, `CtrlRefProvInterface`, and `RefToProvInterface`, and their relations to interfaces via "sets binding from", "gets reference to" and "refers to", respectively, have to be connected with the interfaces of type `Operation` only.

$$(\forall t : \text{CtrlBindReqInterface})\ (t.\text{operation}.\text{type}\ \text{is}\ \text{Operation})$$
$$\wedge\ (\forall t : \text{CtrlRefProvInterface})\ (t.\text{operation}.\text{type}\ \text{is}\ \text{Operation})$$
$$\wedge\ (\forall t : \text{RefToProvInterface})\ (t.\text{operation}.\text{type}\ \text{is}\ \text{Operation})$$

4 A Case Study of a Component-Based System

As a case study, we create a model of a component-based system (CBS) that implements a specific service of a service oriented architecture (SOA). We adopt a specification of the SOA for functional testing of complex safetycritical systems, more specifically *a testing environment of a railway interlocking control system*, which has been described in [7]. The environment allows to distribute and run specific tests over a wide range of different testing environments, varying in their logical position in the system's architecture. The CBS implements service `TestEnvironment`, which executes a test script received from service `TestManager` via its interface `ExecuteTest` and forwards its results back to `TestManager` via interface `asyncReplyET` when the test script is finished. During its execution, the test script is interacting with a tested environment, which is specific to each instance of service `TestEnvironment`[5]. For a detailed description of the SOA and all its services, see [13].

Railway interlocking control systems are safety-critical systems and can be described as component-based systems [4]. A testing environment of such systems has to interact with the systems' components. For that reason, a part of the testing environment, which is directly connected to a system under testing (i.e. the tested environment), has character of a component neighbouring to the system and can be described as the CBS. Moreover, the test scripts are distributed to different instance of service `TestEnvironment`, i.e. different parts of the system's architecture, where they interact with local testing environments. The test scripts act as mobile components in the system's architecture, i.e. in mobile architecture.

[5] Each rail yard has its own instance of the tested environment with specific sensors and actuators where assigned tests are automatically executed.

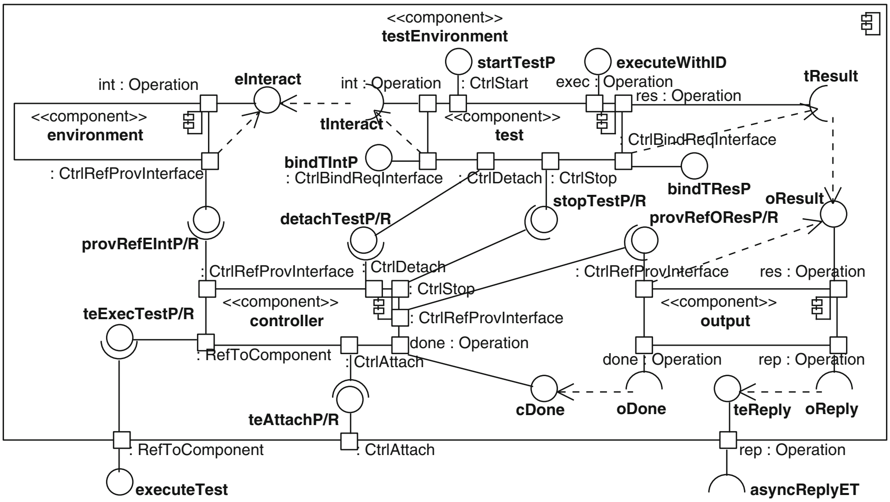


Fig. 5 Composite component TestEnvironment (a specific UML-like notation)

Figure 5 describes composite component testEnvironment, which represents service `TestEnvironment`. The component model (from layer M1) is described as a specific class diagram where entities of the CBS (from layer M0) will be instances of the depicted classes.[6] The class diagram is figured in a specific modification of UML 2 component diagrams' notation.[7]

In Fig. 5, components of the CBS are denoted by UML components, i.e. classes stereotyped as `<<component>>`. Provided interfaces are denoted by UML interfaces, i.e. classes stereotyped as `<<interface>>` realised by related components that own the interfaces, while required interfaces are denoted by UML interfaces used by related components that own the interfaces. A type of a component's interface (see Sect. 3.3) is denoted by an UML port where the port's (optional) name and (mandatory) type are identical to a name and a class of the type of the component's interface. Bindings of functional required and provided interfaces are denoted by UML relations of dependency stereotyped as `<<use>>`. Each binding can have its (optional) name and its type, if needed. In a case of a nameless binding of interfaces, which is common for control interfaces, it is possible to interconnect the interfaces directly. Relations of UML ports of types `CtrlBindReqInterface` or `CtrlRefProvInterface`, which represent control provided interfaces for binding of required functional interfaces or referencing functional provided interfaces, respectively, are denoted by UML relations stereotyped as `<<use>>`.

[6] Another, but less comprehensible, method is to describe the component model as an object diagram where objects in layer M1 are instances of classes from the metamodel in layer M2 and represent abstractions of entities of the CBS from layer M0.

[7] The aim is to simplify the diagram and reuse the well-established UML notation, although it is not formally defined as an UML profile.

Component testEnvironment receives a test script via provided interface executeTest, which is internally processed by component controller. The script is represented by a fresh component, which does required testing after binding of its interfaces to component environment.

At first, component controller attaches the new component as a subcomponent test of component testEnvironment via its control interface teAttachP. Then, it binds interfaces tInteract and tResult of the new component to interface eInteract of component environment and interface oResult of component output, respectively. Finally, component test is activated via interface startTestP and executed with a new identifier via interface executeWithID. The identifier is also returned by component testEnvironment as a reply of the test script's submission.

Component test performs the test script by interacting with component environment via its interface eInteract. When the test script is finished, component test sends the test's results and its identifier to component output via its interface oResult. Then, component output notifies component controller via its interface cDone and forwards the results and the identifier out of the component testEnvironment via its external interface asyncReplyET.

After component controller is notified about the finished test script, it is able to receive and execute another test script, i.e. to attach a new component in the place of component test. Before that, component test with the old script is stopped via interface stopTestP and detached via control interface detachTestP.[8]

5 Conclusion and Future Work

In this paper, we have presented a MOF-based metamodel for modelling of component-based systems with mobile architecture. The proposed metamodel has defined classes for primitive and composite components, the components' functional and control interfaces, specific bindings of the interfaces, and types of their operations. An application of the metamodel has been demonstrated on the case study of the component-based system implementing a specific service for a testing environment of a railway interlocking control system.

We are currently working on an integration of the metamodel into modelling tools based on Eclipse Modeling Framework and on developing a graphical editor in Eclipse Graphical Modeling Framework for structural and behavioural modelling of service-oriented architectures and underlying component-based systems.

Acknowledgements This research was partially supported by the BUT FIT grant FIT-10-S-2 and the Research Plan No. MSM 0021630528 "Security-Oriented Research in Information Technology".

---

[8] In the diagram in Fig. 5, only these two interfaces of test are connected with controller, because the rest of the test's interfaces are used only during its nesting and their connections do not exist outside of controller component.

References

1. Avgeriou P, Guelfi N, Medvidovic N (2004) Software architecture description and UML. In: Jardim Nunes N, Selic B, Silva A, Toval A (eds) UML satellite activities. Lecture notes in computer science, vol 3297. Springer, Berlin, pp 23–32
2. Balasubramaniam D, Morrison R, Oquendo F, Robertson I, Warboys B (2005) Second release of ArchWare ADL. Technical Report D1.7b (and D1.1b), ArchWare Project IST-2001-32360
3. Barros T (2005) Formal specification and verification of distributed component systems. PhD thesis, Universite de Nice—INRIA Sophia Antipolis
4. Bowen JP, Stavridou V (1993) Safety-critical systems, formal methods and standards. IEE/BCS Softw Eng J 8(4):189–209
5. Bruneton E, Coupaye T, Stefani JB (2004) The Fractal component model. Draft of specification, version 2.0-3, The ObjectWeb Consortium
6. Bures T, Hnetynka P, Plasil F. (2006) SOFA 2.0: Balancing advanced features in a hierarchical component model. In: Proceedings of SERA 2006, IEEE Computer Society, Seattle, WA, pp 40–48
7. Donini R, Marrone S, Mazzocca N, Orazzo A, Papa D, Venticinque S (2008) Testing complex safety-critical systems in SOA context. In: CISIS. IEEE Computer Society, Los Alamitos, pp 87–93
8. Medvidovic N, Taylor RN (2000) A classification and comparison framework for software architecture description languages. IEEE Trans Softw Eng 26(1):70–93
9. Meta object facility (MOF) core specification (2006), version 2.0. Document formal/06-01-01, The Object Management Group
10. UML infrastructure, version 2.1.2 (2007). Document formal/2007-11-04, The Object Management Group
11. Oquendo F (2004) π-ADL: an architecture description language based on the higher-order typed π-calculus for specifying dynamic and mobile software architectures. ACM SIGSOFT Softw Eng Notes 29:1–14
12. Oquendo F (2005) UML 2.0 profile for ArchWare ADL Technical Report D1.8, ArchWare Project IST-2001-32360
13. Rychlý M (2010) A case study on behavioural modelling of service-oriented architectures. e-Inf Softw Eng J 4(1):71–87

What Happens Before a Project Starts?—Project Start-up from the Supplier Perspective

Paula Savolainen, June M. Verner, Lesley P. W. Land and Graham C. Low

Abstract Before an outsourced software project officially begins the contracting or supplier organization has already expended effort. Although project start and start-up effort impact on project success in most cases these are undefined concepts. There are no clear definitions of project start, start-up or the activities that should be completed before project start either in the literature or in practice. Ambiguity around project start sets up risks to the profitability of a project and therefore makes the real success of a project not only uncertain but difficult to measure. A vague project start also makes comparisons between projects and between organizations unreliable. In this paper, we describe a pilot study that reviews project start, project start-up, and project start date, and then investigates what the key activities of the supplier are normally performed by the end of the project start-up phase. We use interviews with software supplier practitioners to define those key activities.

1 Introduction

In order to have a prosperous relationship between a customer and a software supplier, their joint projects need to be successful. The ISO/IEC 12207 standard defines a project as an endeavour with defined start and finish dates, undertaken to create a product or service in accordance with specified resources and requirements [14]. The definition of project success depends on the point of view taken and can be defined as: (1) meets planning goals, (2) provides end-user (we use the term "customer") benefits, and (3) provides contractor (we use the term "supplier") benefits (i.e., commercial success of the project and potential for future revenues) [6]. Without a general understanding of all three project success criteria and their implications, it is less likely that real project success will be achieved. The first project success criterion, the ability of the project to meet the planning goals, is closely related to the traditional measures of project success, namely cost, time and quality [2].

P. Savolainen (✉)
School of Computing, University of Eastern Finland, P.O. Box 1627, 70211 Kuopio, Finland
e-mail: paula.m.savolainen@uef.fi

Lero—The Irish Software Engineering Research Centre, University of Limerick, Ireland

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_52,

The second project success criterion, i.e., end user or customer benefits provided by the project, can be defined by the project's impact on general corporate strategy, business operations, research and development, IS/IT development, and facilities provision and management [5]. These benefits cannot always be measured when the project ends because they may take some time to eventuate and it may take years before any actual customer benefits can be estimated. Customer benefits are not the same as having a project delivered on time, within budget and of appropriate quality [5]. There have been studies which show that projects may be successful although they have been clearly over budget and over time [1, 4, 17, 24].

The third project success criterion, supplier benefits, which is important for the research described here, is necessary for a long-term relationship between a supplier and a customer. It is necessary for a supplier to run a profitable business; therefore the overall project portfolio of the supplier needs to make a profit in the long run. Haried and Ranamurthy [11] note that one of the main aims of a supplier is to get additional future business, therefore one of the main criteria for the success of current activities, from the supplier's point of view, is the possibility of future deals with the customer. In order to gain the required benefits, there needs to be a common understanding between the supplier and the customer of the project, its scope, timetable, and costs and these are formed through negotiation between the customer and the supplier.

However, in a recent study it was discovered that what work is included in the project work from the supplier side is not straightforward [21]. Irregulatiries were found from the moment the project was supposed to start. There may be several weeks between a customer order and the point in time when the project is ready to start. During that time a considerable amount of effort may have been made on the project, in addition to effort directly related to start-up activities. We discovered this when we asked "What is 'project start' and how it is defined?" and are in line with [3], who has found that there is a certain minimum effort required in the start-up phase. Therefore, in this study we extended this question to ask:

> What are the key activities of the software project start-up phase that enhances overall project success from a business perspective (i.e. the project supplier perspective)?

In order to answer this question in the next section we present a literature review and discuss previous research. In Sect. 3 we describe interviews with software supplier practitioners who were asked the research question. Section 4 provides our results and Sect. 5 presents our conclusion and a discussion of further work.

2 Literature Review

The aim of our literature review is to gain an understanding of what the literature has to say about project start and project start-up, its importance in projects in general, and the possible activities that might occur during project start-up. The literature we examined includes the existing standards (including ISO, IEEE, and project management), standard software engineering and information systems development texts (e.g. [18, 20, 22]), journals such as The Information Systems Journal, Software Process Improvement, and Project Management Journal, as well as other peer reviewed articles from databases such as the ACM portal, Wiley InterScience database, the IEEE database, Elsevier ScienceDirect and SpringerLink database.

A search through relevant standards (such as ISO, IEEE and project management) using keywords such as "project", "project start", and "project start date" revealed no useful definitions for project start or related terms. We also examined ISO/IEC 12207, ISO/IEC 15288, ISO/IEC 15504, ISO/IEC 16085, and ISO/IEC 16326 [12–16], more closely, without any success. The well known project management standard, "A Guide to the Project Management Body of Knowledge" (PMBOK) does not describe when a project actually starts, or project start, or project start date [19]. However, PMBOK defines an Initiation Process Group, during which the project charter is developed and stakeholders are identified. When the project charter is approved, the project becomes officially authorized. Fangel provides a definition for project start-up [9]:

> Project start-up is a unified and systematic management process which quickly generates a platform for taking off and for getting going effectively.

He also describes the differences between project start and project start-up by using an example from ship-building industry [9]:

> . . . To me it is natural to distinguish between to start and to start-up. When you are going to drive a car, you start by merely turning the key, releasing the clutch, and simply drive away. You rarely give any thought to the matter of performing the kick-off. When you are going to run the diesel engine of a ship, you perform a start-up which is a process involving several activities all needed before the marine engineer can give the final "Go".
> Examples of the activities are the manning of the start-up, communication with the captain, fuel check, lubrication of bearings, starting pumps, initiation of filters, and building up sufficient air pressure. Such a professional start-up process is the basis for getting the engine going, but at the same time it gives an effective and economical operation of the engine. It seems to me that the difference between a project start and a project start-up is just as obvious as the difference between starting a car and starting up a ship's diesel engine.

Using the example above he succeeds in clarifying the difference between project start and project start-up. However, he gives an incomplete definition for project start and project start date [9]:

> The formal project start may be at the beginning of the start-up process, subject to approval of the developed project plans. Alternatively, the start-up process may be partly or fully carried out before the formal project start.

For the supplier company the project start-up phase can be seen a phase beginning from a moment the company has received an order from the customer or the customer has indicated some other way that they will order a software development project from the supplier; it ends when the project has been started. The project start-up phase and its relationship with the sales process and the actual project are depicted in Fig. 1.

In [21], practitioners were interviewed to determine when suppliers considered the actual start of a project. The most common responses were "We got the order" and

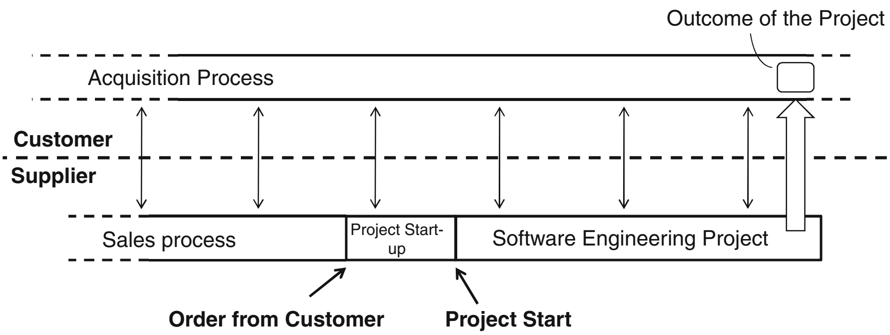


Fig. 1 The start-up phase of a project. [21]

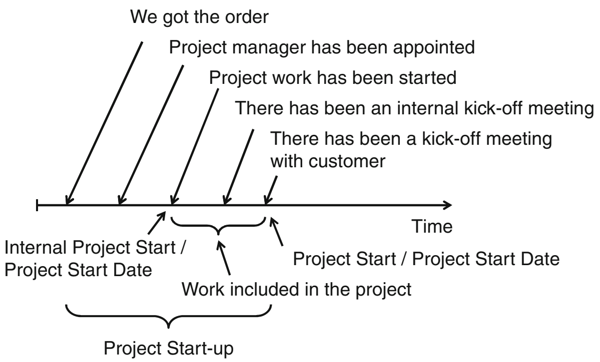


Fig. 2 The definitions of the project start in a timeline. [21]

"Project work has been started"; one interviewee defined the project start via an internal kick-off meeting. It is possible to place the definitions presented in Savolainen [21] within a time-scale and the placement represents the relative ordering of the definitions as shown in Fig. 2.

There may be several weeks between an order and a kick-off meeting with the customer. During that time a considerable amount of effort may have been made on the project, in addition to effort directly related to start-up activities. Therefore definitions for project start, project start date, and project start-up effort included in the project provided in [21] are important. The project start is the day when the supplier and the customer have the project kick-off meeting. This moment is defined also as the project start date. The customer may expect the supplier to have the project team up and running immediately after that meeting. The project start-up activities that are required should be performed before the project start. The start-up activities should be included in the project work from moment when the project work has been started (internal project start/project start date) to the moment when the kick-off meeting has been performed with the customer. In that span a remarkable amount of work has been done and that effort should be included into the project despite the fact that the work is invisible to the customer [21].

Egginton discusses the start-up process and its importance for project success [7]. As a part of his study he formulated a list of start-up activities which are an essential part of large infrastructure projects. Although the project domain is different, his list is quite similar to that required for software development projects. According to Egginton, during the start-up phase a project team is formed, responsibilities assigned, procedures established, tools and controls installed, communications set up and initial contact between the team and the customer made. He emphasizes that the project manager with a just formed project team is faced with a multitude of activities ranging from understanding organizational aspects to developing specifications, from assessing risks of the contract to ordering some part of development from a subcontractor. Egginton recommends that the start-up phase should begin with a handover of project responsibility from the sales organisation to the project organisation. After a successful handover the next step is the rapid launch of the project, which is best achieved with a project kick-off or "start-up" workshop.

Egginton's observations are supported by Barry et al. [3] who have noted that while studying the relationship between software project duration and project effort, that for a given project, a minimum time is required to get complex work started. An organization needs time to set up the project team, train them and allow them to become familiar with the project.

Fangel argues that an appropriate project start-up process should be adopted for each project [8]. He emphasizes that no two projects are identical and every project start-up varies depending on the characteristics of the project [9]. He has identified two essential activities for project start-up: project planning and team building. During project planning the project's objectives are specified and the main project processes and organization are defined. During the team building process, social relationships between the project's participants are created, and rules of cooperation are clarified. Depending on the character of the project, different procedures generate a complete platform of understanding, plans, and cooperation for the effective execution of the project [9].

Turner and Cochrahe agree with Fangel on the need for a customized start-up process [23]. They have judged projects against two parameters: how well defined the goals are, and how well defined the methods are [23]. Like Fangel they argue that depending on the project type, an appropriate project start-up process should be adopted. They suggest that software development projects provide an example of projects where project goals are not well defined but methods are. Therefore in software development projects, the start-up process should focus on defining the purpose and objectives of the project, and converting these into a design of the project outcome which will deliver the required benefits to customer. Thus negotiations within the project team and between the team and the project's sponsor are essential. Once the purpose and the objectives have been defined, the start-up process focuses on refining them, and launching the project with appropriate project team and organization.

PMBOK describes project management through nine different knowledge areas and five project management process groups [19]. One process group is the Initiation Process Group, during which a project charter is developed and stakeholders are identified. When the project charter is approved, the project becomes officially authorized. The project charter is meant to link the project to the ongoing work of the organization and authorize the project. Other activities identified as belonging to the Initiation Process Group are defining the initial scope, committing initial financial resources, selecting the project manager, developing clear descriptions of the project objectives including reasons why a specific project is the best alternative to satisfy requirements, and giving the project manager the authority to apply organizational resources to the subsequent project activities.

The literature therefore gives some indication of what possible project activities may be included in the project start-up phase for non-software projects and some limited insights for software projects, mainly from a project management perspective. A more holistic understanding of project start-up activities is specifically required for software projects, and from the business perspective of software project suppliers. The distinction between the project management and business perspectives may be critical as project success criteria and definitions may have different foci. For example, success criteria from the project management perspective may focus particularly on timely delivery of software with at least the majority of client's core requirements properly implemented. However, from the supplier's business perspective, project success criteria may be shifted to current/future profitability (cost) and team performance issues.

3 Research Methodology

The interviews described in this section are part of a larger study which included two different sets of interviews performed in four software engineering companies. The larger study aims to gain a better understanding of those activities which are performed in a supplier company before the project has been started and affect the project during its life-cycle. One set of interviews concentrated on activities performed in the tendering process and another set concentrated on the initiation activities performed in the project start-up phase. This study concentrates on the interviews which discussed the project start-up phase.

Two of the software engineering companies where interviews were performed were involved in software development projects for various customers. The other two companies were involved with embedded software projects with close cooperation with industrial companies. The number of employees varied from 20 to 230 employees and the age of the companies varied from 5 to 23 years at the time the interviews were performed. The size of the project groups varied from 2 to 10 team members, the duration of the projects were mainly from 1 to 15 months although one company had projects which lasted for up to 3 years. The main characteristic of all four participant companies is that they deliver unique products (software or embedded software, or in some cases specialized hardware with embedded software) for their customers. For these companies projects are their main way of doing business.

The practitioners interviewed were selected by the higher-level management of the companies who were asked to select project managers or other people responsible for project management for interviews. The interviewees included eleven Project

Managers, one Business Unit Manager, one Team Manager, and one Engineering Manager; altogether 14 practitioners were interviewed. We use title "project manager" when referring to all interviewees irrespective of their titles. No other people except the project managers were interviewed.

An interview instrument consisting of main themes and a form for background data were developed. The interview instrument was constructed by one researcher and validated by two other researchers. The interviews were semi-structured, and took the form of a discussion, using the interview instrument as a guide. Every interview was recorded and the recordings were transcribed to text. The transcribed texts were then analysed. All interviews were reviewed and a summary of each interview (that was a general description of the interviewee's experience of project start-up and activities done during the start-up phase) was developed. We concentrated on extracting information that describes what ought to be the key activities that will enhance overall project success from the project manager's point of view. From these descriptions each relevant activity was selected, listed, and analysed. Activities with similar meanings were grouped and named as a key activity. Analysis was then performed and the results are described in the next section.

4 Results

One of our main findings is that project managers find themselves in many different situations. Some project managers first found out about the project only when they were nominated as the project manager for the project. However, one project manager acted as a salesman, sold the project, got the deal, gathered as good a project group as was possible, and continued with the project as a project manager. There are many differences between project manager responsibilities and what their actual contributions are to the future project *before* there is a project at all. This result was independent of the title of the interviewees.

Key activities which arose from interviews were:

1. The project manager should formulate a "big picture" of the project and its objectives;
2. The project team should understand what they are going to do; and
3. The customer and the supplier should gain a common understanding of the project and its objectives.

The first key activity, the project managers' need to gain a holistic understanding of the project and its objectives, were clearly seen in the interviews. As the interviews took the form of discussions, guided by the interview instrument, it was remarkable, that 10 project managers out of the 14 emphasized the importance of this holistic understanding for project success. Some project managers mentioned the need for a better understanding of the customer's business. Additionally, they mentioned that they should better understand the customer's other information systems and the role of this project and its relationship with their other information systems. This information cannot normally be found in the project's documentation. The project manager needs to understand the purpose of the project because without that understanding it is not possible to write the project plan or rewrite any project plan already created during the tendering phase. This applies especially to the schedule because only with a holistic understanding it is possible to estimate if the original schedule is realistic or not.

Without this understanding the project should not be allowed to start. This was clarified by one project manager who said:

> The customer had a tight schedule and we needed three—four designers and to work with existing specifications. When looking back we should not have done any design work on the first day. Instead, I should have moved to the customer for a while to fathom out, what was the need and if the specifications were finished or not.

Some project managers commented that at this stage, it is not possible to understand in detail what will be done during a project but a more general understanding will first have to be gained. It is essential to understand early what is really important so that the project team is able to devote its time to important matters instead of less important details. Of the four project managers who did not emphasize the importance of an early understanding of the project and its objectives, one was responsible for selling the project to the customer so was mainly concerned with how to make the project team understand what they are going to start to do. This is discussed later in this section.

Just as a project manager needs to gain a holistic understanding of the project and its objectives, the whole project team also needs to understand what they are going to do. One project manager described this point as follows:

> It's better that we don't do anything but sit on our hands for two weeks so that it's clear to everyone what are we aiming for and what are the goals.

The most often mentioned practice to contribute to an understanding of the project within the project team was the reading of available documents. An internal meeting was also mentioned by some project managers who used that meeting as an informative meeting. Some project managers, during the project start-up phase, utilized project planning meetings with the project team as a learning situation; while participating in project planning the project team has an opportunity to understand the scope and objectives of the project itself, and can provide valuable contributions to the project. Besides project planning meetings within the supplier company, meetings with customers were seen also as valuable learning situations. During customer meetings the project team familiarizes itself with business processes from customers side, and thereby it becomes possible for the whole project team to recognize critical parts of the project before project start.

The project managers did not rely only on the information about the project that existed in their own company alone. They understood that before the project was ready to start that there should be a common understanding of the project and its goals between the supplier and the customer, which was the third key activity. Nine project managers out of the 14 mentioned this key activity but it is not always clear that this common understanding exists before project start. While the supplier company has created its own vision of the project this can be quite different from the customer's vision.

> The customer images they will get more and the supplier images they will do less

was noted by one of the project managers. However, another project manager said that the customer's needs could have changed after the supplier company submitted the tender. One reason for changed needs is that it sometimes takes a surprisingly long time between the customer's decision to order the project from the supplier, and the project proceeding to the start-up phase. Therefore the current situation must be checked. By ensuring a common understanding with the customer before project start the project managers can avoid encountering problems in the future. There will be changes during the project execution, and this was expressed by one project manager:

> We shouldn't sustain a daydream at the beginning of the project it would be clear what the final outcome is to be.

If the customer and the supplier have a common understanding of the project and its objectives before project start, it is possible to mitigate any problems connected to change management and negotiations on how to compensate the changes. It was noticed that the project managers had a realistic attitude that there will be changes during project execution and they must prepare for future changes during the startup phase.

5 Conclusions and Future Work

The work presented here discusses project start, project start-up, and what activities are included in the project start-up. Although the amount of effort spent in the startup activities may be remarkable, the relationship between those activities and the project start has not been clearly defined in the relevant literature. Therefore we analyzed interviews conducted during a larger study and identified the following key activities:

- Project managers need to gain a holistic understanding of the project and its objectives. They need to have a good understanding of the customer's business, the customer's other information systems and the role of this project and its relationship with their other information systems. It is essential they understand early what is really important so that they know where to focus their attention.
- The project team needs to understand what they are going to start to do by reading available documents. An internal meeting can be used to provide information. Some project managers utilize project planning meetings with the project team as a learning device; while participating in project planning the project team will better understand the scope and objectives of the project itself, and can provide valuable contributions to the project.
- The customer and the supplier need to develop a common understanding of the project and its objectives and not only rely on information that exists in their own company alone. A common understanding of the project and its goals should exist between supplier and customer.

Although the study was based on a limited number of interviews its results can be considered valid because similar problems have been reported elsewhere [10]. Further work includes investigation of the start-up phase more closely, especially in supplier companies. We should also consider the start-up phase and its connections to both successful and failed projects.

Acknowledgements This research was funded by the Finnish Funding Agency for Technology and Innovation (Tekes) with the grant 70011/08, and supported in part by Science Foundation Ireland grant 03/CE2/I303_1 to Lero—The Irish Software Engineering Research Centre.

References

1. Agarwal N, Rathod U (2006) Defining 'success' for software projects: an explaratory revelation. Int J Project Manag 24(4):358–370
2. Atkinson R (1999) Project management: cost, time and quality, two best guesses and a phenomenon, it's time to accept other success criteria. Int J Project Manag 17(6):337–342
3. Barry EJ, Mukhopadhyay T, Slaughter SA (2002) Software project duration and effort: an empirical study. Inf Technol Manag 3(1–2):113–136
4. Berntsson-Svensson R, Aurum A (2006) Successful software project and products: an empirical investigation. In: Proceedings of the 2006 ACM/IEEE international symposium on empirical software engineering, ISESE 2006. ACM, New York, pp 144–153
5. Cooke-Davies T (2002) The "real" success factors on projects. Int J Project Manag 20(3):185–190
6. Dvir D, Raz T, Shenhar AJ (2003) An empirical analysis of the relationship between project planning and project success. Int J Project Manag 21(2):89–95
7. Egginton B (1996) The project start-up process—getting it to work better. Eng Manag J 6(2):88–92
8. Fangel M (1984) Planning project start-up. Int J Project Manag 2(4):242–245
9. Fangel M (1991) To start or to start-up? That is the key question of project initiation. Int J Project Manag 9(1):5–9
10. Haapio T, Ahonen JJ (2006) A case study on the success of introducing general non-construction activities for project management and planning improvement. In: Proceedings of PROFES 2006, LNCS 4034. Springer, New York, pp 151–165
11. Haried P, Ramamurthy K (2009) Evaluating the success in international sourcing of information technology projects: the need for a relational client-vendor approach. Project Manag J 40(3):56–71
12. ISO/IEC (ed) (2004) ISO/IEC 15504: information technology—process assessment. ISO/IEC, Geneva
13. ISO/IEC (ed) (2006) ISO/IEC 16085: systems and software engineering—life cycle processes—risk management. ISO/IEC, Geneva
14. ISO/IEC (ed) (2008) ISO/IEC 12207-2008: systems and software engineering—software life cycle processes. ISO/IEC, Geneva
15. ISO/IEC (ed) (2008) ISO/IEC 15288-2008: systems and software engineering—system life cycle processes. ISO/IEC, Geneva
16. ISO/IEC (ed) (2009) ISO/IEC 16326: systems and software engineering—life cycle processes—project management. ISO/IEC, Geneva. Final Draft
17. Linberg KL (1999) Software developer perceptions about software project failure: a case study. J Syst Softw 49(2–3):177–192
18. Pressman RS (2005) Software engineering: a practitioner's approach, 6th edn. McGraw-Hill, New York

19. Project Management Institute (ed) (2008) A guide to the project management body of knowledge, 4th edn. Project Management Institute, Newtown Square, PA. ANSI/PMI 99-001-2008
20. Royce W (1998) Software project management: a unified framework. Addison-Wesley Longman, Inc., Reading, MA
21. Savolainen P (2010) Vague project start makes project success of outsourced software development projects uncertain. In: PROFES2010: the 11th international conference on product focused software development and process improvement, LNCS 6156, Springer, New York, pp 351–365
22. Sommerville I (2007) Software engineering, 8th edn. Person Education Limited, Upper Saddle River, NJ
23. Turner JR, Cochrane RA (1993) Goals-and-methods matrix: coping with projects with ill defined goals and/or methods of achieving them. Int J Project Manag 11(2):93–102
24. de Wit A (1988) Measurement of project success. Int J Project Manag 6(3):164–170. doi:10.1016/0263-7863(88)90043-9

Fragmento: Advanced Process Fragment Library

David Schumm, Dimka Karastoyanova, Frank Leymann and Steve Strauch

Abstract Reuse is a common discipline for decreasing software development time and for improving overall quality, independent from the domain. As business processes represent a fundamental asset of an organization, several concepts for enabling reuse during process modeling have been proposed. However, only few concrete examples for reusable process artifacts have been discussed so far. In this paper, we present the concept of process fragments and an example collection of process fragments for illustrating our reuse concept and for showing that it can actually be applied in practice for an easier and faster development of process-based applications. The fragment examples demonstrate different characteristics such fragments may exhibit. We also argue that this work will encourage reuse of process logic in terms of fragments, since it also provides an opportunity to design and develop a process fragment library for collecting process logic explicitly. As technical enabler for the approach we present a prototype called Fragmento.

1 Introduction

The current reusable granules in process design are language constructs like activities, control and data connectors, routing gateways, business rules, variables and other basic artifacts. The next larger and established reusability-related concept is subprocess, which already represents a self-contained and functionally complete unit

D. Schumm · D. Karastoyanova · F. Leymann · S. Strauch (✉)
Institute of Architecture of Application Systems, University of Stuttgart,
Universitätsstraße 38, 70569 Stuttgart, Germany
e-mail: Schumm@iaas.uni-stuttgart.de

D. Karastoyanova
e-mail: Karastoyanova@iaas.uni-stuttgart.de

F. Leymann
e-mail: Leymann@iaas.uni-stuttgart.de

S. Strauch
e-mail: Strauch@iaas.uni-stuttgart.de

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_53,

for modeling and execution. Next in size are process templates, process variants and reference models, which are already the largest granular units for reuse and cover reusability and customizability for whole processes. Reuse of only a part or an artifact of a process is not covered by these approaches. We argue that within the range from basic language constructs to sub-processes and process templates there needs to be another, smaller unit of reuse which should allow fine-grained reuse of business logic. Process fragment presents a concept that fills this gap. The concept for reuse that we propose allows an easier and faster development of process-based applications. This includes for instance applications based on Web service compositions. Besides, it is a widely spread trend in software engineering to partially separate the application functions from the process logic orchestrating them.

This paper describes a collection of reusable building blocks for usage in process design based on the concept *process fragment*. In addition, we present an infrastructure component enabling the storage and management of fragments, which we call process fragment library. We discuss concrete examples of process fragments which we have identified during our research. The fragments have specific characteristics in which they differ from each other, for example the number of exits or if constraints are imposed on them. One objective of this work is to show the usefulness of the concept of process fragments by providing real examples that one can actually work with. In other disciplines, such as in traditional programming, code fragment libraries are a quite common source for reuse. Also in areas that are not related to computer science, libraries of reusable building blocks are widely used, e.g. in chemistry [1]. Within computer science such libraries are sometimes referred to as repository [2]. Various works on these special purpose databases exist for instance in the field of semantics in business processes [3] or agent systems [4]. In order to advance the state of the art we advocate the use of a repository providing advanced functionality for managing reusable process artifacts in different process languages. Additionally, we want to encourage the identification and publication of further process fragments in order to create an open library for capturing the progress of research and development in this field and see the presented work as technical enabler.

This paper is organized as follows: Sect. 2 describes a general concept for process fragments and exemplifies domain-specific extensions of this concept. Section 3 describes several process fragment examples. In Sect. 4, we present a process fragment library as supporting infrastructure for our approach. Related work on concrete process fragments is presented in Sect. 5. Finally, Sect. 6 summarizes the paper and identifies future work.

2 The Concept of Process Fragments for Reuse

In this section, we present briefly the concept of process fragments that is independent from the platform and technology that is chosen for the actual process language, implementation and serialization format for process fragments. In [5] we have given a general definition for the notion of process fragments. “A process fragment is defined as a connected graph with significantly relaxed completeness and consistency criteria compared to an executable process graph. A process fragment is made up of activities, activity placeholders (so-called regions) and control edges that define control dependency among them. A process fragment may but is not required to: define a context (e.g., variables) and contain a process start or process end node. It may contain multiple incoming and outgoing control edges for integration into a process or with other process fragments. A process fragment has to consist of at least one activity and there must be a way to complete it to an executable process graph. Therefore, a process fragment is not necessarily directly executable and it may be partially undefined [5]". Depending on the particular language that is chosen for implementation, further characteristics are conceivable such as explicit data flow represented by data edges. Based on this definition we are able to express reusable pieces of process structure without limiting the expressiveness to single entry single exit (SESE) structures.

Depending on the particular application domain further requirements on the characteristics of a process fragment might be necessary. For instance, the focus in our research is set on managing compliance [5]. Compliance refers to all measures that need to be taken in order to adhere to laws, regulations and internal policies (corporate guidelines within the company). It is required that process fragments must not be changed to ensure the corresponding compliance feature. This means that the fragment may only be used the way it has been designed and only particular parts of it may be changed. This way it can be better ensured that after integration of a fragment into a process it still implements the compliance requirement that it has been designed for. For the usage of process fragments in the field of compliance we proposed in [5] additional characteristics: (1) a process fragment may be parameterizable in order to mark points of variability which can render it abstract. The fragment is completed (i.e. concretized) when incorporated into a process; (2) the placeholders contained in a process fragment (i.e. regions) may be constrainable. By constraining the regions it can be defined how those placeholders may be filled with activities or other fragments.

Process fragments are reusable in process design in general and also in the field of (Web) service composition in particular. Apart from applying the fragments in modeling compositions they can be used to specify additional information like usage scenarios associated to services, compliance criteria a service meets etc.

3 Process Fragment Collection

We use the Business Process Modeling Notation (BPMN) [6] for representing process fragments graphically. We extended this notion with a cloud icon for representing a region. Parameters and constraints are expressed with an annotation icon that we created, see Fig. 1. Entries and exits of a fragment are represented by control links that either have no target or no source. We use this notation in the scope of this work to ease understanding. The code fragment specified in the Business Process

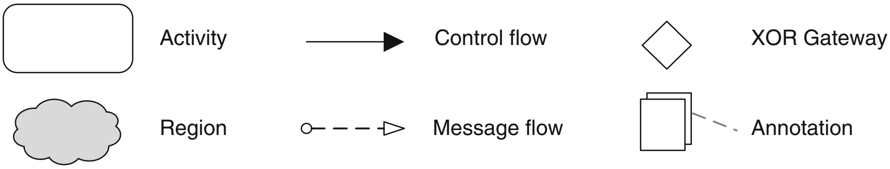


Fig. 1 Process fragment constructs

Execution Language (BPEL) [7] we discuss here does not make use of the extensions and can be represented with native language constructs.

As mentioned in the introduction, the fragments discussed in the following sections have specific characteristics in which they differ from each other. The first example fragment (approval) has multiple exits. The next fragment extends the fragment for approval with constraints and regions. The fragment realizing the "4-eyes principle" also uses constraints and contains a region for customization. A symmetric counterpart for usage in choreographies is provided by the fragment for secured interaction. The fragment for trusted timestamp exemplifies a domain-specific fragment for reuse. A particular control structure is implemented by the fragment for avoidance of infinite waits. These characteristics can be exploited as classification schema to support efficient search in the fragment library. The fragments of our collection have been manually identified in an industrial case study (compliance in a loan originating process) which has been defined by our project partner Thales Services SAS, France. For this paper we have selected rather simple fragments of this case study as they clearly illustrate the key concepts of our approach. After identification we have modeled the fragments in a process design tool and stored them in our process fragment library for later reuse in process modeling.

3.1 Process Fragment for Approval

In many business processes and also in workflows (i.e. in the technical implementation of a process [8]) a step for checking a particular situation is required. For instance, for quality assurance there needs to be a check for mistakes and also for authorization reasons checks are necessary, as discussed in [5]. Typically, there are even multiple approvals within a single process, e.g. in approval chains. Figure 2 shows the process fragment for approval in BPMN. This fragment is applicable in almost any process language though. The fragment states that if a certain condition is met, a particular situation is assessed. This fragment has a single entry and in our design it has multiple exits, one for acceptance and the other for rejection. It has some characteristics which are likely to be parameterized: These are the activation conditions in which cases this approval needs to be performed, a staff query or a Web service for performing the check, and respective input values that should be approved, e.g. a document. Those parameters need to be set during concretization, i.e. during integration of the fragment into a process.

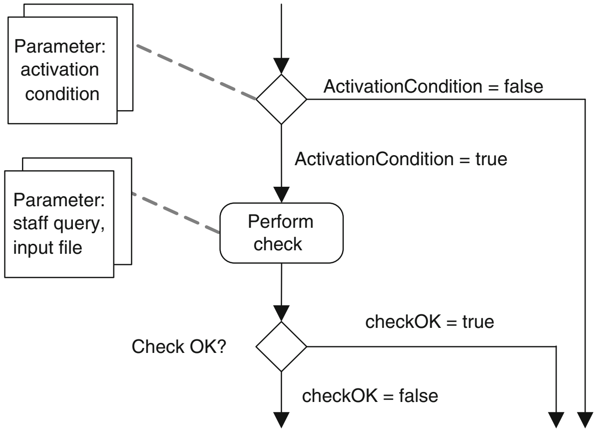


Fig. 2 Process fragment for approval

3.2 *Process Fragment for Approval with Constrained Region*

In Sect. 2 we have remarked that specific extensions for process fragments depending on the application domain can be useful. In the field of compliance one required characteristic is that only particular parts of the fragment may be changed during integration into a process. For this we use an annotation mechanism for describing how particular parts may be changed during customization and integration of the corresponding fragment into a process, see Fig. 3.

To allow modification of the inner structure of a fragment in a controlled manner we propose to impose constraints on regions for compliant composition. In other scenarios regions could also be used without any constraints. In this example fragment

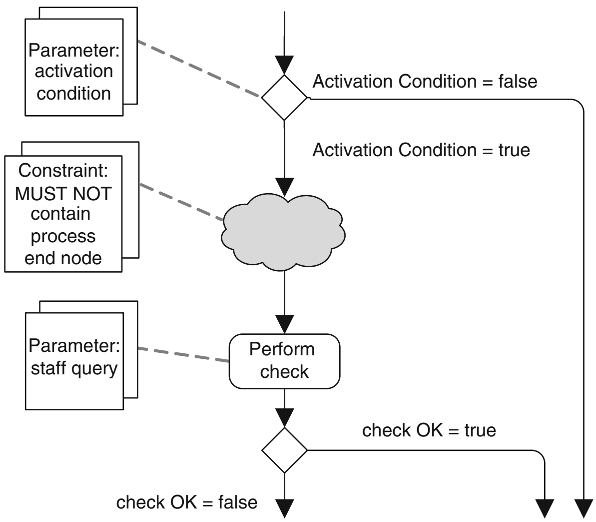


Fig. 3 Process fragment for approval with constrained region

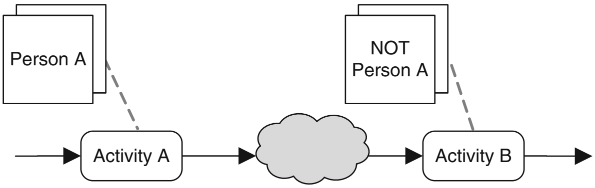


Fig. 4 Process fragment for 4-eyes principle

the region allows integrating other steps in between the entry of the fragment and the approval step. However, disabling the approval must not be possible. Constrained regions could also be placed at the entries and exits of a fragment for enabling a constrained integration into a process.

3.3 Process Fragment for 4-Eyes Principle

The 4-eyes principle (also called segregation of duty) is a frequent compliance requirement used for avoiding misuse and fraud, for security reasons or for avoidance of conflict of interest. For instance, in a banking application the customer requesting a loan and the clerk who may grant it must not be the same person. Typically, this requirement is realized using an annotation mechanism as the fragment in Fig. 4 illustrates, combined with checking during runtime. The fragment in Fig. 4 is designed for sequential execution. For parallel execution or for execution without control dependency other variants of this fragment need to be defined.

3.4 Process Fragment for Secured Interaction

There are many different ways, methods and technologies for making an interaction with a process partner secure. This includes for instance transport layer security, message encryption and usage of signatures. The common way for integration of such functionalities is to annotate the activities which shall be executed in a secured manner. This annotation is interpreted and accordingly executed by the corresponding middleware. Nonetheless it can also be directly integrated into a process. Although the fragment shown in Fig. 5 might only be used for documentation purposes and not be applied in process execution languages due to the before mentioned current practice, it is still an illustrative example for a fragment that has a corresponding counterpart. A counterpart in this context is another fragment designed for interaction and integration with the fragment from the partner's point of view. The number of counterparts depends on the particular interaction scenario. These kinds of fragments are important in Web-based application integration in which multiple processes and services need to interact with each other in a well-defined manner.

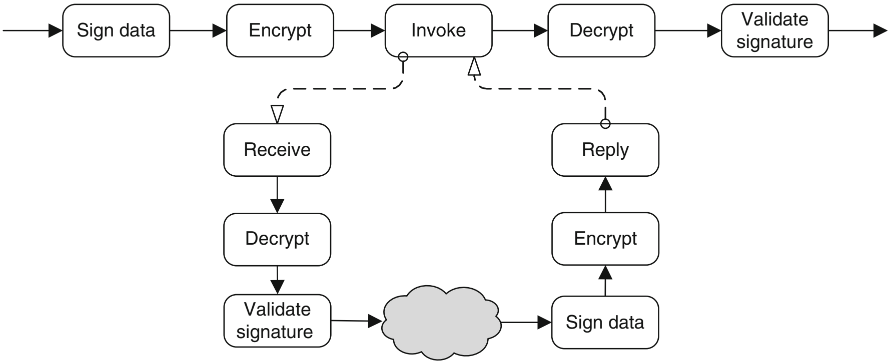


Fig. 5 Process fragment for secured interaction (*upper part*) and the symmetric counterpart (*lower part*)

3.5 Process Fragment for Trusted Timestamp

For some business processes it is necessary to store a timestamp, e.g., when an offer is being sent out to a customer. For compliance reasons this timestamp needs to be "trusted" in particular cases, this means it has to be issued by a certified timestamp provider. Figure 6 shows a process fragment that has been designed according to the procedure for retrieving and validating a trusted timestamp defined in [9]. Basically, this fragment could be used for integration of trusted timestamps into a process without requiring in-depth knowledge on the details of the procedure. Possibly, this fragment could even be offered from the timestamp provider for easier and faster integration with the offered (Web) services. We argue that process fragments can be used as an annotation to a service (or process) for providing additional meta-information about it, going beyond the description of its interfaces and usage policies. We consider this approach in our process fragment library. Please note that this fragment could also be implemented as a sub-process, however with limited customization capabilities.

3.6 Process Fragment for Avoidance of Infinite Waits

Process fragments are concrete solutions to frequently occurring, but also to specific problems. The process fragment for avoidance of infinite waits (see Listing 1)

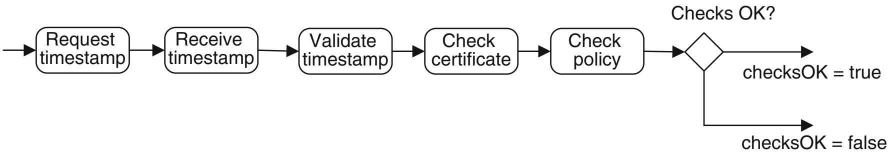


Fig. 6 Process fragment for trusted timestamp

implements a control structure in BPEL which takes care that a process does not hang up in case a service which has been invoked does not respond. This can be achieved by using a `<pick>` construct in combination with the receiving activity `<onMessage>` that awaits the response. If the response is not received in time, the `<onAlarm>` construct registers a timeout and cancels waiting for the message and thereby prevents the process from hanging up. This control structure is not really complex, but in case a process designer is not sure how to deal with this problem it becomes quite handy. Even if the designer knows how to model this, reusing this fragment can at least speed up overall development time. Another fragment defining control structures is best-practice in process design: for dynamically changing endpoint references of service invocations during runtime an `<eventHandler>` construct with a nested `<assign>` activity can be used.

Listing 1. Process fragment for avoidance of infinite waits

```
<sequence name="main">
   <invoke name="invokeService" .../>
   <pick name="pick">
        <onMessage ...>
          <assign name="assignResponse" validate="no" />
        </onMessage>
        <onAlarm for="P1DT00H">
          <assign name="assignTimeoutOccurred" />
        </onAlarm>
   </pick>
</sequence>
```

4 Supporting Infrastructure

In this section we present the process fragment library which is the special purpose component for storage and management of process fragments. We have developed a prototype of such a library, called Fragmento [10], its conceptual architecture is presented in Fig. 7. Fragmento is dedicated to the management of BPEL processes, WSDL documents, WS-Policy Annotations, especially BPEL process fragments and other process-related artifacts.

Beyond the basic functionality for management of versions, locks and (typed) relations we have implemented several functions which are helpful particularly in the management of processes and process fragments, see Fig. 7. For example, the basic search functionality operates on fragment meta-data, like the fragment name, keywords, the number of entries and exits, in which domain it is used and other classifications (currently full text search). In addition, Fragmento provides an extensibility mechanism for integration of custom query functions. This allows the implementation of search functions beyond the meta-data of an artifact, e.g. concerning the structure of a fragment or related to properties of its annotations.

Process design and process enactment require valid models for proper execution. Thus, Fragmento also provides XML schema validation and an extensibility mechanism for additional validation functions that could be used to check if a process model

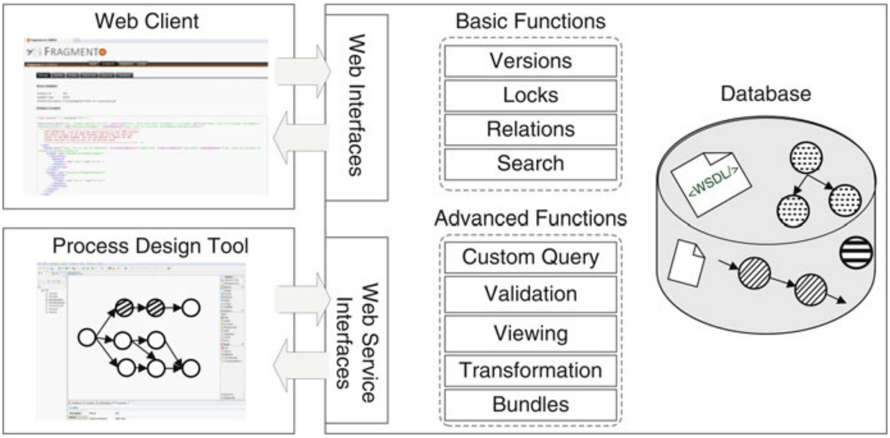


Fig. 7 Conceptual architecture of Fragmento

contains cycles for example. For flexibly creating user-specific representations and variants of processes and process fragments on the fly, Fragmento supports process view transformations [11]. These transformations include for instance the omission of attributes and activities that match particular characteristics (e.g. for removal of activities related to debugging). Furthermore we have integrated a transformation that changes language extensions used in a process fragment (e.g. for representing regions) into standard constructs for compatibility with other tools. We consider the mechanisms for easy retrieval of process fragment information a valuable feature and we therefore support the definition of bundles, which enables packaging all artifacts related to a process (or fragment) into one SOAP message (or Web page).

Fragmento exposes all of its offered functionality as Web service (currently via SOAP/HTTP binding). It also provides a Web-based interface to allow direct access to the repository over the Web. For the Web client we use double tab navigation. On top level the user can choose between the management of artifacts, relations or locks. On the second level the particular management functions for the corresponding top level selection are shown. The integration with a process design tool based on Eclipse is part of our current research agenda.

Fragmento extends an existing repository code base that has been developed by the MASTER project [12]. The technology stack for Fragmento consists of a Tomcat application server which is hosting the repository application. Hibernate is used as data abstraction layer. Furthermore, the Spring Framework is employed for object lifecycle management and a PostgreSQL database is utilized for storage. For the development of the Web service interfaces Axis 2 libraries have been used. The Web client is built using Java Server Pages (JSP) and Tag Libraries for the view, while Servlets are used as controller for handling client requests. Fragmento is a Java application. All the fragments presented in this work can be stored and managed by Fragmento. More documentation of the functionality, provided interfaces, screenshots and implementation details can be found in [10].

5 Related Work

In the following we discuss work related to concrete process fragments. A comparison of our concept to approaches on process reuse in general and a discussion of the life cycle of process fragments can be found in our former work [5]. Notable for this paper are the results of the ProWAP project. With the term Workflow Activity Pattern the authors refer to the description of recurrent business functions that are frequently used in business processes. In [13] a set of seven activity patterns based on literature study is discussed. These activity patterns are namely Approval, Question-answer, Uni-/Bi-directional Performative, Information Request, Notification and Decision Making. In this work the activity patterns are defined as SESE fragments (without placeholders), similar to sub-processes. This limitation intends to ease pattern implementation, pattern reuse and pattern composition within process design tools. We see the patterns discussed in [13] as an additional source of concrete fragments, however the fragments we presented and the patterns discussed in the mentioned article are just the tip of the iceberg.

We would like to stress that we see a difference between the terms pattern and fragment. For instance, in [14] Workflow Patterns are discussed. Workflow Patterns describe elementary language constructs which are supported by workflow languages, for example sequential execution, parallel split or exclusive choice. The expressiveness of workflow languages differs, thus some workflow patterns might be supported by a particular language while some others might not. The patterns described in [14] are somehow reusable building blocks and [15] even shows that these workflow patterns can in fact be applied as modeling granules for accelerating process development. However, a pattern is an abstract solution concept to frequent problems while a fragment is a more concrete solution, possibly to a quite specific problem. A fragment could more or less be compared to a code snippet, while a pattern is more conceptual, like a design pattern in terms of [16]. Another example for patterns in this context is Message Exchange Patterns (MEP) [17]. According to [18], MEPs define the sequence, the cardinality, the source and the recipient of messages. For instance, Request-Response is such a pattern. These patterns can also be applied in process design in the way shown in [15], but still they are quite abstract forms of reusable building blocks.

An approach that provides patterns that enhance the reliability of a BPEL process is shown in [19]. The work makes use of a guideline for defining reusable fault handling logic [20] and discusses four abstract solution patterns and BPEL code fragments for fault-tolerant service invocation. The authors propose to annotate the reliability requirements to the process and use a model transformation to automatically integrate the fragments accordingly. In summary, the fragments described in [19] are domain-specific, concrete, language-specific and, which is most important here, they are reusable and useful for making process design easier and faster.

6 Conclusion and Future Work

In this paper we have presented our approach of process fragments for reuse in process design. The main contribution of this paper is a collection of concrete process fragments which illustrates that there is a need for this concept and that it actually can be applied in practice. The fragment examples we presented are of a rather simple nature in order to clearly illustrate the key concepts of the approach. As technical enabler we presented a process library prototype called Fragmento. The architecture of the prototype and its functionality were discussed in detail. The process library supports storage and management of recurring and reusable process fragments without focusing on a particular application domain.

In many different fields, for instance in grid computing, manufacturing workflows or scientific workflows, there are most likely domain-specific and language-specific but also general process fragments which can be manually identified and subsequently reused. Fragments from particular application domains may also be useful in other domains, or bring up new ideas which are helpful in many fields. In [21] techniques for fragment discovery in the field of scientific workflows have been proposed, they are basically also applicable for fragment discovery in a business context. Furthermore the case study evaluation in [21] states that there is definitely a need for workflow fragments and reusable service composition in e-Science scenarios.

At present, we are investigating methods and limitations of translating fragments representations from one process language into another. We are also investigating techniques for isolating and extracting reusable process fragments from existing processes. In our future research we will work on a classification of the different forms and characteristics of process fragments. We are convinced that diversity in research on fragments will be beneficial for the further development of the overall fundamentals, concepts and related techniques. The presented collection of process fragments can be seen as a starting point for future research concerning reusable building blocks of process logic.

Acknowledgements The work published in this paper was partially funded by the COMPAS project (www.compas-ict.eu) under the EU 7th Framework Programme Information and Communication Technologies (contract no. FP7-215175) and the S-Cube project (www.s-cube-network.eu) under the Network of Excellence (contract no. FP7-215483).

References

1. Maybridge (2007) The maybridge Ro3 fragment library, product flyer. http://www.maybridge.com/images/pdfs/ro3frag.pdf
2. Bernstein P, Dayal U (1994) An overview of repository technology. In: Proceedings of the 20th international conference on Very Large Data Bases (VLDB), Morgan, Kaufmann
3. Ma Z, Wetzstein B, Anicic D, Heymans S, Leymann F (2007) Semantic business process repository. In: Proceedings of the Workshop on Semantic Business Process and Product Lifecycle Management (SBPM)

4. Seidita V, Cossentino M, Gaglio S (2006) A repository of fragments for agent systems design. In: Proceedings of the Workshop on Objects and Agents (WOA06)
5. Schumm D, Leymann F, Ma Z, Scheibler T, Strauch S (2010) Integrating compliance into business processes: process fragments as reusable compliance controls. In: Proceedings of the Multikonferenz Wirtschaftsinformatik (MKWI'10)
6. BPMN (2008) Object Management Group: Business Process Modeling Notation (BPMN), OMG available specification, Version 1.1, Jan 2008
7. BPEL (2007) Organization for the Advancement of Structured Information Standards (OASIS): Business Process Execution Language 2.0 (BPEL)
8. Leymann F, Roller D (2000) Production workflow: concepts and techniques. Prentice Hall, Upper Saddle River
9. Network Working Group (2001) Internet X.509 public key infrastructure time-stamp protocol (TSP), RFC 3161. http://tools.ietf.org/html/rfc3161
10. Fragmento (2010) Fragment-oriented repository. Online documentation. http://www.iaas.uni-stuttgart.de/forschung/projects/fragmento/start.htm
11. Schumm D, Leymann F, Streule A (2010) Process viewing patterns. In: Proceedings of the 14th IEEE international EDOC conference (EDOC 2010). IEEE Computer Society Press
12. European Project MASTER (2009) Technical architecture and APIs for single trust domain. Project Deliverable D2.3.1. http://www.master-fp7.eu
13. Thom L, Reichert M, Iochpe C (2009) Activity patterns in process-aware information systems: basic concepts and empirical evidence. Int J Bus Process Integr Manag (IJBPIM) 4(2):93–110
14. van der Aalst W, ter Hofstede A, Kiepuszewski B, Barros A (2003) Workflow patterns. Distribut Parall Databases 14(1):5–51 (Springer)
15. Gschwind T, Koehler J, Wong J (2008) Applying patterns during business process modeling. In: Proceedings of the 6th International Conference on Business Process Management (BPM). Springer
16. Gamma E, Helm R, Johnson R (1995) Design patterns. Elements of reusable object-oriented software. Addison-Wesley, Boston
17. W3C (2007) W3C Web Services Description Language (WSDL) Version 2.0: additional MEPs, W3C working group note, June 2007
18. Nitzsche J, van Lessen T, Leymann F (2008) WSDL 2.0 message exchange patterns: limitations and opportunities. In: Proceedings of the 3rd International Conference on Internet and Web Applications and Services (ICIW), IEEE
19. Liu A, Li Q, Huang L, Xiao M (2007) A declarative approach to enhancing the reliability of BPEL processes, IEEE International Conference on Web Services (ICWS), IEEE Computer Society
20. Zeng L, Lei H, Jeng J, Chung J, Benatallah B (2005) Policy-driven exception-management for composite web services. In: Proceedings of the 7th IEEE International Conference on E-Commerce Technology (CEC), IEEE Computer Society
21. Goderis A, Sattler U, Lord P, Goble C (2005) Seven bottlenecks to workflow reuse and repurposing. In: Proceedings of the 4th International Semantic Web Conference. Springer

Improving the Cold-Start Problem in User Task Automation by Using Models at Runtime

Estefanía Serral, Pedro Valderas and Vicente Pelechano

Abstract Automating user behaviour is one of the most important challenges in Ambient Intelligence. Most of the proposed approaches to confront this are based on machine-learning algorithms. Using them, user routine tasks are inferred from user past actions and then automated when needed. However, these approaches present the cold-start problem, i.e. they cannot infer routine tasks until they gather sufficient user actions. We improve this problem by using a modelling approach. We propose two models to specify the routine tasks known at design time and a software infrastructure that automates them when needed by interpreting these models at runtime. Thus, we achieve to automate user routine tasks from the very beginning. Furthermore, we complement this infrastructure with a model-based API that allows these models to be modified at runtime. Thus, we provide a high-level repository for user behaviour information and an initial background to improve behaviour predictions.

1 Introduction

Ambient Intelligence (AmI) [1] is a computer paradigm that tries to make real the vision of Mark Weiser [2] where environments are electronically enriched to make them sensitive to user needs and anticipatory of user behaviour. This anticipatory aspect implies that the system is able to adapt to user behaviour in order to automatically perform specific tasks just in the moment users need to perform them. To deal with this, machine-learning approaches (e.g. MavHome [3], PlaceLab [4], Neural Network House [5], etc.) have done an excellent work automating user routine tasks inferring them from user past actions by using prediction algorithms. For instance, these approaches can learn that the alarm clock always goes off at 8:00 a.m., and then the lights are switched on, and the heating of the bathroom is switched on; and then can automate the learnt actions. However, these approaches present an important problem in the initial deployment of the system: the cold-start problem.

E. Serral (✉)
Universidad Politécnica de Valencia, Valencia, Spain
e-mail: eserral@pros.upv.es

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_54,

By *cold-start* problem we refer to the fact that the system cannot infer user actions until it gathers sufficient user past actions. It is well-known that prediction algorithms require many training examples to be accurate [6, 7]. This implies that Am I systems do not provide one of their intrinsic characteristics such as anticipation until a long time after it is deployed. For instance, in the MavHome Project [3] and in the PlaceLab [4], authors state that they need a long period for gathering enough data to properly predict actions, which may take weeks or even months.

In this work, we improve this problem by using a model-driven approach that makes use of models at runtime. We base on the fact that many user tasks to be automated can be known at design time. For instance, we can know if users want that the system goes off the burglar alarm and calls the police when an intruder gets into our home; or if users want that the system lowers every blind and winds up every awning when it starts to rain. Thus, we propose to capture this user behaviour at design time in order to constitute a high-level repository of user behaviour information that can be used to achieve two main goals: (1) to automate users tasks from the early deployment of the system and (2) to provide machine-learning approaches with an initial knowledge base that can be used to improve their behaviour predictions. To achieve this, we present a context-adaptive task model that allows user behaviour to be described through a hierarchical composition of tasks. These tasks are specified according to context [8] allowing us to describe behaviour that must be automated when context conditions are satisfied. This context information is described in an ontology-based context model. We use the concept of task because it has proved to be effective in user behaviour modelling [9, 10].

Both the task model and the context model are brought a step further being also used at runtime. We present a software infrastructure that is capable of automating the specified user tasks when needed by interpreting the models at runtime. Furthermore, we complement this infrastructure with an API that can be used to: (1) update the models at runtime, which allows maintaining the initial repository of user behaviour updated in time, and (2) facilitate the inference of user actions at runtime by using concepts of a high level of abstraction (the same used to create the models).

Therefore, the contributions of this work are the following:

1. A model-based technique for describing context-adaptive tasks that allows us to define an initial automation plan for AmI systems.
2. A software infrastructure that allows us to use context-adaptive task models and context models at runtime in order to develop AmI systems that are capable of automating user behaviour as soon as the system is deployed.
3. An API, which can be used by third approaches, to use task models and context models as a user behaviour repository that facilitates the inference of user actions.

The remainder of the paper is organized as follows: Section 2 presents some background and related work. Section 3 gives an overview of our approach. Section 4 describes the context and task models. Section 5 introduces the API to manage these models at runtime. Section 6 presents the infrastructure to automate user behaviour. Section 7 introduces a discussion about our work. Section 8 describes the evaluation performed. Finally, Section 9 explains the conclusions and further work.

2 The Cold-Start Problem in AmI Systems

The cold-start problem is a term originally used in recommendation systems [11]. It concerns the issue that systems cannot draw any inferences for users about which it has not yet gathered sufficient information. This term can also be applied to indicate the initial learning time that systems require for automating user behaviour.

Although not coined by any term, the need for large data sets to perform accurate predictions is a problem historically identified in the general area of prediction algorithms [7]. To deal with this, we base on the strategy of using some data examples that are close to the data that must be predicted (see for instance [12]). Thus, we propose a data repository based on models, in which examples of behaviour to be automated are described. These examples can be taken as a base for further predictions as we discuss in Sect. 7. Moreover, we go a step beyond and these examples are not only used with prediction purposes but they represent real automation requirements of users that are taken into account from the very beginning.

With regard to current approaches for user behaviour automation in AmI systems, three of the most important approaches that use prediction algorithms are MavHome [3], PlaceLab [4], and Neural Network House [5]. However, although all of them acknowledge that an initial period of training is required, there is little work that focus on improving this drawback. Other solutions such as rule-based approaches have been also proposed to confront the challenge of automating user behaviour. These approaches programme rules to automate user actions when certain context condition arises. Some examples are the works proposed in [13] and [14]. These approaches do not have the cold-start problem since a training period is not required. However, they are not appropriate for automating user complex tasks and generally require large numbers of rules which have to be manually programmed [15]. In addition, they do not adapt the automated user actions after system deployment.

3 An Overview of the Approach

To improve the cold-start problem of user task automation in AmI system, we propose a software infrastructure were models coexist with code at runtime. The main elements of this infrastructure are the following (see Fig. 1):

- **Pervasive Services:** an AmI system provides users with pervasive services that control the environment devices (e.g. switch lights on, activate the security alarm, etc.) and sense context information (e.g. presence detection, temperature, etc). The implementation of these services is out of the scope of this work. Any implementation strategy can be used. In particular, we have used a model-driven development (MDD) method, which was presented in [16], to automatically generate the Java/OSGi code [17] of these services from a set of models.

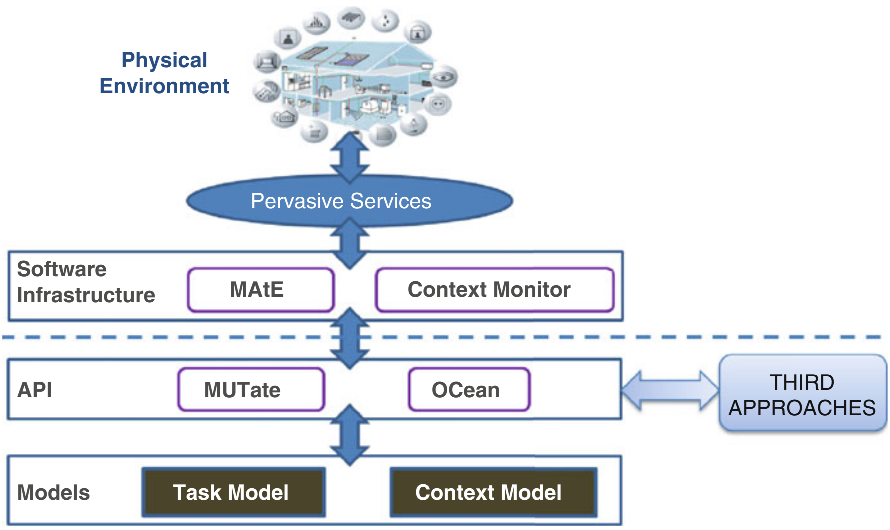


Fig. 1 Overview of the soft ware infrastructure

- **Models:** the context model and the task model describe the context of the system and the behaviour that must be automated. These models are created at design time and directly interpreted at runtime, without any kind of transformation.
- **Software Infrastructure:** it is in charge of integrating models with pervasive services. It has two main elements: (1) a **context monitor** that updates the context model according to context changes, and (2) the Model-based user task Automation Engine (**MAtE**) that interprets the models and executes the corresponding behaviour (pervasive services) in the appropriate context.
- **APIs:** two APIs are provided in order to manage models at runtime: (1) the Model-based User Task management mechanisms (**MUTate**), which provide constructors to manage the task model and (2) the Ontology-based Context model management mechanisms (**OCean**), which provide constructors to manage the context model. The two APIs are used by MAtE and the context monitor to interact with the models. Note, however, that these two APIs can also be used by third approaches in order to use the proposed models at runtime.

4 Modelling of User Behaviour

To specify the user behaviour that must be automated, we base on the behaviour pattern concept. A *behaviour pattern* is a set of tasks habitually performed when similar contexts arise [18]. Thus, we propose two models: a context model, which specifies the context on which the behaviour patterns depend, and a task model, which specifies the behaviour patterns according to the context of the context model.

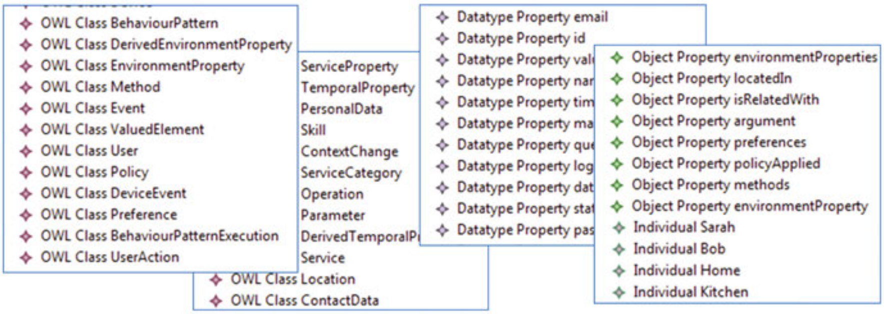


Fig. 2 Example of a context model

The **context model** is defined by using a context ontology. Several studies, such as [19, 20], state that the use of ontologies to model context is one of the best choices to guarantee a high degree of expressiveness and semantic richness. Ontologies also exhibit prominent advantages for reasoning and reusing context as well as making the integration of environments easier. They also provide a formal analysis of the domain knowledge, and allow common understanding of the structure of contexts to be shared among users, devices, and services. For these reasons, we defined an ontology to support the specification of context information.

To create the context model, we use the Ontology Web Language (OWL) [21]. OWL is an ontology markup language that greatly facilitates knowledge automated reasoning and is a W3C standard. Thus, the classes of the ontology are defined by OWL classes, and the context specific of the system is defined by OWL individuals, which are instances of these classes. Figure 2 shows an example of the information captured by this ontology. This figure shows classes such as *User, Location, EnvironmentProperty,* etc. More information about this model can be found in [16].

The **context-adaptive task model** allows the behaviour patterns to be specified by splitting them up into simpler tasks whose execution adapt to context. The proposed context-adaptive task model is based on the *Hierarchical Task Analysis (HTA) technique* [22], which breaks down tasks hierarchically into other tasks. Thus, analysts can start by representing more abstract tasks that describe what users initially want in a general way, and then extract more specific tasks. We propose defining a task hierarchy for each behaviour pattern. The root task represents the behaviour pattern and has an associated context situation, which defines the context conditions whose fulfilment enables the execution of the behaviour pattern. It has also a priority (High, Medium, and Low) to establish the priority of execution of the pattern in case several patterns are enabled at the same time. This root task can be broken down into *Composite Tasks* (which are intermediate tasks) and/or *System Tasks* (which are leaf tasks). The Composite Tasks are used for grouping subtasks that share a common behaviour or goal. The System Tasks are related with a pervasive service whose execution carries out the task. Both types of task can have a context precondition, which defines the context situation that must be fulfilled so that a task is performed (if the precondition is not fulfilled, the task will not be executed). These tasks also inherit the context preconditions of their parent task. In addition, each task is defined by a Task name and an internal Task ID. As examples, Fig. 3 shows the modelling of the *GoingOut* and *StormSecurity* patterns using our task model.

In order to break down a behaviour pattern or a composite task into simpler tasks we propose two task refinements: the exclusive refinement (represented by a solid line) and the temporal refinement (represented by a broken line). The exclusive refinement decomposes a task into a set of subtasks, in such a way that only one subtask will be executed (disabling the others). For instance, in the *GoingOut* pattern, the *manage home resources* task is refined in two tasks by exclusive refinements in such a way that only one of these tasks will be executed (depending on whether it is holiday or not). The temporal refinement also decomposes a task into a set of subtasks; however, this refinement provides constraints for ordering the execution of these subtasks. This is known in HTA as a plan [22]. Using this refinement, the subtasks of the same parent can be linked using temporal operators. The temporal operator between two tasks will also be applied to their child tasks. For reasons of brevity, we only present the operators that are shown in Fig. 3. To define them, we based on [9], which provides one of the richest set of temporal operators:

- T1>>T2, enabling: the *T2* task is triggered when the *T1* task finishes. For instance, the *notify user to activate security* task of the *GoingOut* pattern is only triggered when the *manage home resources* task has been performed.
- T1(i)>>T2, enabling with information passing: when the *T1* task is finished, the information *i* is transferred to the *T2* task. For instance, when the *calculate rainfall* task of the *StormSecurity* pattern finishes, it passes the amount of rainfall quantity to the *update irrigation timetable* task that is then enabled.
- T1|||T2, concurrent tasks: the *T1* and *T2* tasks can be performed in any order. For instance, the execution order of the *lower blinds* and *wind up awnings* tasks of the *StormSecurity* pattern is not relevant.

In addition, to properly capture the AmI system automation requirements, we add the following operator:

- T1>>[c]>>T2, enabling when the c is fulfilled: after executing T1, T2 is enabled when the condition *c* is satisfied. *c* can be either a temporal restriction (e.g. 3 min after the *wind up awnings* task finishes, the *switch sprinklers off* task is enabled) or a context condition (e.g. after the sprinklers have been switched off, when it stops raining, the *modify irrigation system task* is enabled).

To specify the context conditions (in context situations, task preconditions, and relationships), we use a logical expression. This expression combines any number of basic expressions linkedby the logical connectives: and (AND), or (OR), not (NOT), equalities (=), inequalities (!=), and more (>) or less than (<). The variables used in these expressions have to be previously specified in the context model.

Using Models at Runtime. The context model is created by means of the EODM plugin of the eclipse platform [23]. This plugin allows an OWL model to be visualized and edited using a tree graphic editor as well as using OWL code. Fig. 2 shows an example of context model shown in a tree representation. To be able to use this model at runtime, we use its representation in OWL code. The task model is specified by means of a modelling tool developed by using the Eclipse platform and the EMF and GMF plugins [23]. This tool allows the model to be graphically edited (as shown in Fig. 3), and also stored in XMI (XML Metadata Interchange). To be able to use this model at runtime, we use its representation in XMI.

5 An API for the Management of Models at Runtime

In this section, we present the APIs developed to interact with the models at runtime: OCean and MUTate.

5.1 OCean

Context is captured in the context model as OWL individuals. Thus, we have implemented a set of Ontology-based Context model management mechanisms (OCean) to manage individuals in our context model. OCean provides an API that allows any individual of the context model to be created, obtained, modified, and deleted. Specifically, this API consists of a *ContextModel* class that allows us to open the context model, save it, and manage its individuals (*addIndividual, getIndividual* and *deleteIndividual*) in a generic way. The API also provides a Java class for each one of the OWL classes defined in the context ontology. Each Java class provides:

- An attribute for each one of the properties and relationships of its OWL class; e.g., the *User* class has *DNI* and *preference* as attributes.
- Get, set and delete methods for each one of these attributes; e.g., *getDNI*.
- An add method for the attributes whose type is a List. This method allows an element to be directly added to the list; e.g., *addPreference* method.
- Get and delete methods for the attributes whose type is a List. These methods allow us to update or delete an element of the list by searching for it using one of its identifier properties; e.g. *getPreferenceByName*.

We have used the OWL API 2.1.1 [24], SPARQL [25] and the Pellet reasoner 1.5.2. [26] to implement OCean. A partial view of the source code of OCean is shown in Fig. 4. As an example, the figure shows the *deleteIndividual* method of the *ContextModel* class. Using the OWL API, this method first creates a remover entity, it then gets the individual that is to be deleted, and passes it to the remover entity for deletion. Finally, the method applies the changes in the context model.

5.2 MUTate

In order to manage task models at runtime, we have developed a set of Model-based User Task management mechanisms (MUTate). MUTate provides an API that

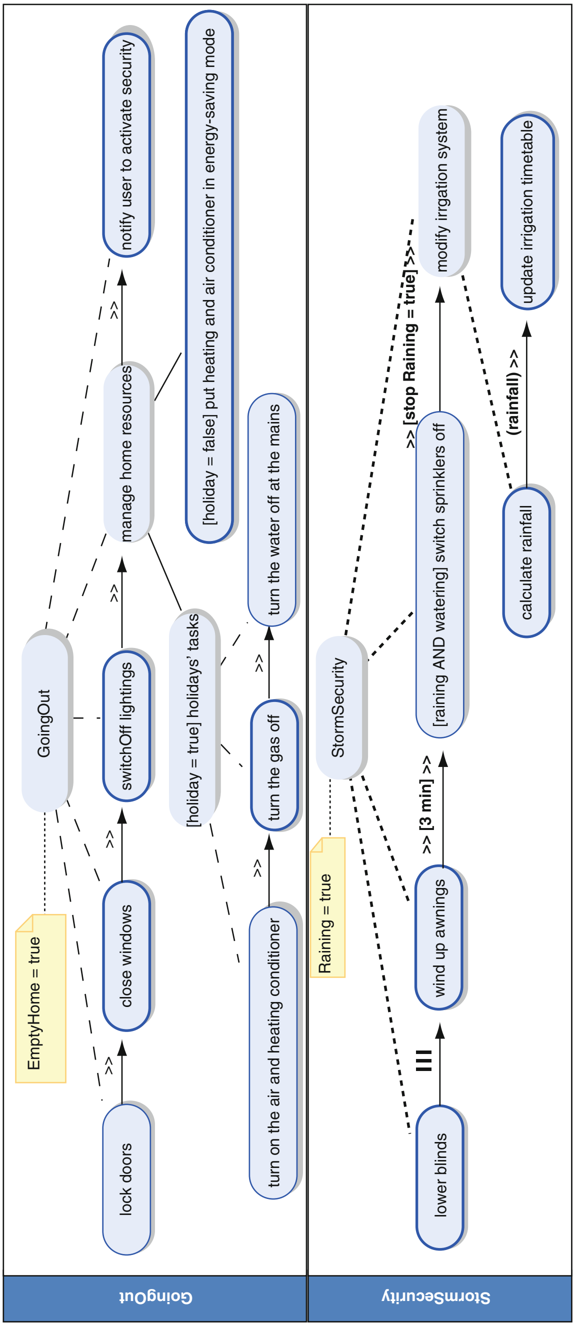


Fig. 3 Examples of behaviour pattern modelling

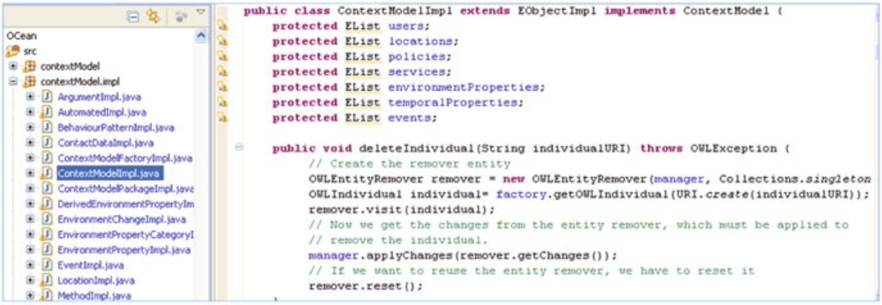


Fig. 4 Code example for context model management

allows any elements of the specified task model (such as Behaviour Patterns, Tasks, Relationships, etc.), to be created, obtained, modified, or deleted. Specifically, this API provides a *Factory* class for creating new instances (of the classes defined in the task metamodel) of a task model. The API also provides a Java class for each one of the classes of the task model metamodel. Each class provides:

- An attribute for each one of the properties and relationships of the metamodel element that the class represents; e.g., the *BehaviourPattern* class has *name* and *task* as attributes.
- Get, set and delete methods for each one of these attributes; e.g., getName.
- An add method for the attributes whose type is a List. This method allows an element to be directly added to the list; e.g., *addTask* method.
- Get and delete methods for access to a certain element of the attributes whose type is a list. These methods allow us to get and delete an element of the list by searching for it by using one of its identifier properties; e.g. *getTaskByID*.

We have used the EMF, EMF Model Query (EMFMQ), and EMF Model Transaction (EMFMT) plugins of the Eclipse Platform to implement MUTate. A partial view of the source code of MUTate is shown in Fig. 5. As an example, the figure shows the *getBehaviourPatternByContextSituation* method of the *TaskModel* class.

This method returns the pattern whose context situation is the same than the BPContextSituation argument value. To find the pattern, it searches for it by using a query statement built with EMFMQ.

6 A Software Infrastructure for Automating User Behaviour Patterns

In this section, we introduce a software infrastructure that allows the defined automation plan to be executed by interpreting the task model and the context model at runtime. To achieve this, this infrastructure uses the APIs explained in Sect. 5. This

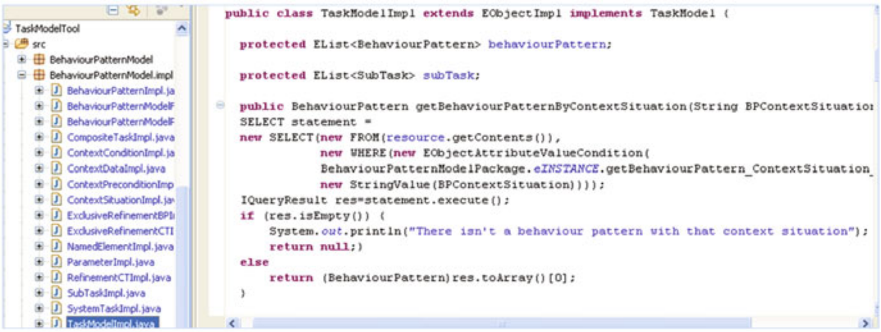


Fig. 5 Code example for task model management

infrastructure (see Fig. 6) is defined by two main elements implemented by using Java/OSGi technology: a Context Monitor and MAtE.

The context monitor captures and processes context changes, and updates the context model accordingly. Note that these changes are physically detected by sensors, which are controlled by pervasive services. Thus, to capture context changes, the monitor is continuously monitoring the execution of the pervasive services (step 1). When a context change is detected, the context monitor reflects the change in the context model by using OCean (step 2). Next, the context monitor informs **MAtE** (the Model-based user behaviour pattern Automation Engine) about the context that has been updated (step 3). Then, MAtE performs the following actions:

1. It analyzes the task model in order to check if there is a context situation that depends on the updated context information (step 4). For instance, if presence inside the home is no longer detected by the presence services, the *EmptyHome* context property is updated by the context monitor. This property is used in the *GoingOut* situation. Then, MAtE checks whether or not the new value of this property make the context situation to be fulfilled. To do that, MUTate is used.
2. When a context situation is fulfilled, the behaviour pattern related with it is executed (step 5). To do this, MAtE executes the system tasks of the corresponding

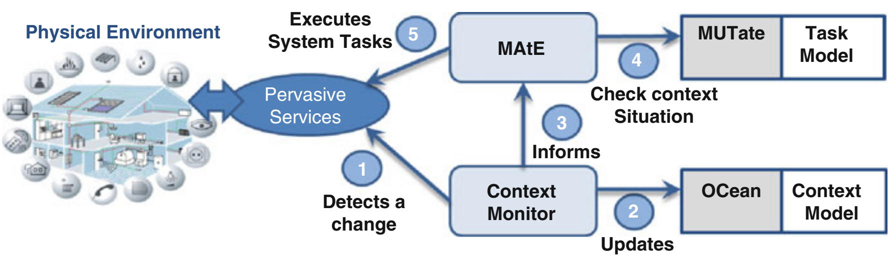


Fig. 6 Automating user behaviour patterns

pattern according to the temporal relationships specified among them. For instance, if the *GoingOut* context situation is fulfilled on a working day (see Fig. 3), the system closes all the windows and doors of the house, switches off the lights, and puts the heating and air conditioner in energy-saving mode (note that the tasks to be performed on holidays are disabled). Finally, the system sends a notification to the users to remind them to activate the home security. To execute the System tasks MAtE executes the service associated to each system task according to the established temporal relationships.

7 Integrating Task and Context Models with Prediction Algorithms

Our work can be divided into two independent parts (see Fig. 1): (1) the models and the API to manage them at runtime, which can be used by third approaches and (2) our infrastructure made up of MAtE and the context monitor, which use the models and the API to automate user behaviour patterns. Next, we explain how these two parts of our work can be integrated with prediction algorithms.

On the one hand, third approaches that use prediction algorithms to automate user behaviour can benefit from our approach by using the proposed models as an initial knowledge base. First, this knowledge base, created at design time, describes a behaviour that users need to be automated. This can help third approaches to improve the cold-start problem and start to automate actions since the system deployment. In order to access this initial knowledge base, third approaches can use the MUTate and OCean APIs. Second, the knowledge base defined by models can help prediction algorithms to improve the analysis of user behaviour, since they have initial data to make predictions. In addition, the models allow this analysis to be performed using high-level concepts such as *task, location, user, context condition,* and so on, instead of analysing the raw data captured by sensors. Finally, note that MUTate and Ocean APIs can also be used to update the proposed models at runtime, which allow third approaches to use them as a high level user behaviour repository.

On the other hand, our approach can be improved by complementing it with a prediction algorithm. Note that user behaviour may change after a specific period of time and the defined behaviour patterns may become useless for users. To avoid this, these patterns have to be evolved to adapt to new user needs. In this evolution, prediction algorithms can play a main role.

8 Evaluation

MUTate and OCean manipulates models at runtime. This is subject to the same efficiency requirements as the rest of the system because the execution of model operations impacts overall system performance. Therefore, these operations have to be efficient enough so that the system response is not drastically affected. Thus, we performed an experiment to quantify the temporal cost of the operations of MUTate and OCean that access models. To test these operations we used the context model and an empty task model to be randomly populated by means of an iterative process. The context model was populated with 100 new context values each iteration, while the task model was populated with one new pattern whose task structure formed a perfect binary tree, varying its depth and the width of the first level each iteration.

After each iteration, we tested all the model operations 20 times and calculated the average temporal cost of each one. The operations with the highest temporal cost that access the task model were the operations for getting, updating or deleting a task. These costs were very similar since all of them make the same query to obtain the corresponding task. Even with a model population of 45612 tasks, these operations provided a fast response (< 250 ms). The operation with the highest temporal cost that access the context model was the *checkCondition* operation, which took 9 ms with 100 context values of the consulted classes and 11 ms with 6000 context values. These results show that the response time is not drastically affected as the size of the models grows.

9 Conclusions and Further Work

We have presented a model-driven approach that use models at runtime in order to solve the cold-start problem of user behaviour automation in AmI systems. In particular, we have proposed a context-adaptive task model and a context model that provide a rich expressivity to specify behaviour patterns. We have also designed and implemented a software infrastructure to support the automation of these behaviour patterns since the system deployment. To achieve this, the infrastructure interprets the models at runtime. In addition, we propose a set of APIs that allow third approaches to use these models as a repository of user behaviour that facilitates the inference of behaviour patterns by using high-level concepts.

As further work, we plan to develop an end-user tool that provides user-friendly interfaces to allow end-users to easily change at runtime the behaviour patterns that are automated. These interfaces will use MUTate and OCean to update the models according to the changes. We are studying the development of user interfaces inspired by the *Natural Programming* and *Visual Programming* end-user approaches [27]. We also plan to combine this end-user tool with a prediction algorithm. This algorithm will be used to automatically infer new behaviour patterns from the user action history. The end-user tool will periodically show users the inferred behaviour patterns so that users can modify and add the patterns to the system if they so desire.

Acknowledgements This work has been developed with the support of MICINN under the project SESAMO TIN2007-62894 and co-financed with ERDF, in the grants' program FPU.

References

1. ISTAG (2001) Scenarios for ambient intelligence in 2010
2. Weiser M (1991) The computer of the 21st century. Scient Am 265:94–104
3. Youngblood GM, Cook DJ, Holder LB (2005) Managing adaptive versatile environments. Pervasive Mobile Comput 1:373–403
4. Intille SS, Larson K, Tapia EM et al (2006) Using a live-in laboratory for ubiquitous computing research. In Proceeding of Pervasive
5. Mozer MC (1998) The neural network house: an environment that adapts to its inhabi-tants. In AAAI Spring Symposium on Intelligent Environments
6. Webb GI, Pazzani MJ, Billsus D (2001) Machine learning for user modeling user. Modeling User-Adapted Interact 11:19–29
7. Valiant LG (1984) A theory of the learnable. Commun ACM 27:1134–1142
8. Dey K (2001) Understanding and using context. Personal Ubiquitous Comput 5:4–7
9. Paternó F (2002) ConcurTaskTrees: an engineered approach to model-based design of interactive systems
10. Masuoka R, Parsia B, Labrou Y (2003) Task computingthe semantic web meets pervasive computing. In The Semantic WebISWC
11. Schein A, Popescul LH, Ungar MJ et al (2002) Methods and metrics for cold-start recommendations. In ACM SIGIR Conference
12. Billsus D, Pazzani M (1999) A hybrid user model for news story classification, in user modeling (Proceedings of the Seventh International Conference)
13. García-Herranz M, Haya PA, Esquivel A et al (2008) Easing the smart home: semi-automatic adaptation in perceptive environments. JUCS 14:1529–1544
14. Henricksen K, Indulska J, Rakotonirainy A (2006) Using context and preferences to implement self-adapting pervasive computing applications. Sofware: Pract Exp 36:11–12
15. Cook DJ, Das SK (2005) Smart environments: technologies, protocols, and applications. John Wily, New York
16. Serral E, Valderas P, Pelechano V (2010) Towards the Model Driven Development of context aware pervasive systems. Pervasive Mobile Comput 6:161–180
17. OSGI, http://www.osgi.org/
18. Neal DT, Wood W (2007) Automaticity in situ: the nature of habit in daily life. In: Bargh JA, Gollwitzer P, Morsella E (eds) Psychology of action: mechanisms of human action. Oxford University Press, London
19. Chen H, Finin T, Joshi A. An ontology for context-aware pervasive computing environments. Special Issue on Ontologies for Distributed Systems, KER, pp 197–207
20. Baldauf M, Dustdar S, Rosenberg F (2007) A survey on context-aware systems. Int J Ad Hoc Ubiquitous Comput 2:263–277
21. Smith MK, Welty C, McGuinness DL (2004) OWL Web ontology language guide, W3C Recommendation, http://www.w3.org/TR/owl-guide/
22. Shepherd A (2001) Hierarchical task analysis. Taylor & Francis, London
23. Eclipse Platform, www.eclipse.org
24. Horridge M, Bechhofer S, Noppens O (2007) Igniting the OWL 1.1 Touch Paper: The OWL API, OWLED
25. SPARQL Query Language (2008) http://www.w3.org/TR/rdf-sparql-query/
26. Sirin E, Parsia B, Grau BC et al (2007) Pellet: a practical OWL-DL reasoner. J Web Semantics 5:51–53
27. Pérez F, Valderas P (2009) Allowing end-users to actively participate within the elicitation of pervasive system requirements through immediate visualization. in Rev

Integrated Approach in Management and Design of Modern Web-Based Services

Jaroslav Škrabálek, Lucia Tokárová, Jiří Slabý and Tomáš Pitner

Abstract The new kind of applications being currently developed deserves more attention in the user interface development area. This paper takes into account slightly different needs of new applications such as Web 2.0 and considers new approaches in their development cycle. Namely, the development is more based on user comfortability needs and the paper stirs the processes needed to accomplish this successfully and early in the development phase. Thus, minimizing the bucket demands which is critical for project managers. The paper outlines also future steps to reach even better results.

1 Introduction

The importance of modern web-based services has arisen in the field of software development as well as in ICT industry in general. Since then, significant changes in approach to a software product can be seen. It does not matter whether we consider classical desktop application or web application. But thanks to web applications, the process of changes has been catalysed and user is becoming the primary factor determining the product's resulting form—both from the point of functionality and user interface.

Web technology advancements caused transition of companies from desktop applications to the Internet services. During this period, processes of the web transformation have been replaced in favour of the higher level of Internet development. The Internet from the Web 2.0 point of view is not a provider of exclusively static, passive information anymore. The Internet became a standalone platform offering *Rick Internet Applications* (RIA)—modern web-based services [12]. Increasing complexity and usability of RIA have changed people's understanding not only of the Internet but the whole development process of such application, communication interface of that application and the approach to people who create the application in

J. Škrabálek (✉)
Faculty of Informatics, Masaryk University, Brno, Czech Republic
e-mail: skrabalek@fi.muni.cz

J. Pokorny et al. (eds.), *Information Systems Development,*
DOI 10.1007/978-1-4419-9790-6_55,

the end. These significant questions regarding the software development [14] are now broadly discussed and the process itself has been redefined. The most important change is diverting from purely technological perception of a final software product to the consideration of product's user acceptance. Not the functionality is standing on the first place but usability. This change has second major influence on developers. Companies gradually turn their task orientation to developers where they are carriers of the tasks and potentially of a product success. The more a developer is satisfied the more the task will be fulfilled in a satisfactory manner.

2 Combining Disparate Technical and Humanistic Aspects in IT Project Management

Development of modern web-based services is very appropriate for an application in new humanistic management approaches with outlook emphasizing common human needs. It is because they are strongly oriented to the fundamental social principles of their users. A social aspect for the modern web is essential. Whereas the content of Web 2.0 is mainly user-generated, communities and collaboration are highlighted. After a very short period, new Web 2.0 phenomena capture the heart of millions Internet users. They are blogging, using wikies (which become modern encyclopedias), various community networks such as MySpace (approximately 200 million users and slightly decreasing), Facebook (about 400 million users and growing), LinkedIn, XING, etc. One of the core concepts of Web 2.0 became so called *Folksonomy*—attaching unstructured—a combination of "folk" and "taxonomy". This concept was defined thanks to long interest of Thomas Vander Wal (he observes a phenomenon of tagging since the pre-web era in 1980s). Folksonomy represents a process of tagging with social titling (social indexing, social classification, collaborative tagging) of resources. Tagging is very similar to the way how our brain works. It creates transcendent manifold associations rather than inflexible categories. More plainly—folksonomies are sets of names tagged to the content by users without any fixed predetermination. It is usually an informal free form.

In these days, human management is characterised by acceptance of business enterprise as anthropomorphized, being where persons in the enterprise define the culture and quality of the organization and its products as well.

2.1 Management

Organizations should build up a unity to achieve the community of persons being stronger as a community [8].

The business cannot be considered only as a "society of capital goods" because it is also a "society of persons" [13]. Managers have to motivate people around them to acquire virtues and try to discover and promote beliefs and values within the organizational culture that foster human virtue, in all its forms, to its fullest extent. Humanity management is neither a naive approach nor a lack of realism. It fulfills aim of the third approach to humanistic management, which is still emerging. It considers a team as a real community of persons. The connection to virtual community of Web 2.0 is obvious [9]. This humanistic project management approach is a real challenge in order to achieve higher moral quality in management, more efficient development and in the end more efficient organizations. Only by this the full potential of the organization, mainly formed by people, will be exposed. Possible aspects of person (developer) psychology development can be conceived as a continuum encompassing training, conditioning, introjection and receptive learning of foreign experiences. Personal values and project goal are on the same level of importance and they get in the relationship between project manager and developer. In that manner the growth process is created and sense of personal and project development is higher.

2.2 Person-Centered Approach

Thanks to close connection of Web 2.0 to the *Person-Centered Approack* (PCA) via Web 2.0 definition basis (e.g. collaborations, user-generated content, and collective intelligence), it is an application of Web 2.0 technologies that supplements original intention of PCA properly [20]. PCA was introduced by an American psychologist Carl Rogers. According to Rogers, the essentials are three attitudinal conditions [17]. They are a motive power in personal development of each person. These conditions are:

- acceptance (unconditional positive regard),
- emphatic understanding and
- congruence.

Congruence expresses personal integrity composing from experiences and feelings in mutual complementation. Integrity comprises acceptance and emphatic understanding as well. Those attitudes are beneficial for all relationships characterized by psychical development—relationship between child and parents, partners, or relationship between manager and developer too. We called this *Developer-Centered Approack* (DCA). With the help of changes that Web 2.0 introduces to the software development, the classical software development and project management methodology adapts especially in methodology exactly for the Internet environment. In a certain point of view, the PCA alters the management methodology as well (Fig. 1).

2.3 Methodology

In order to choose proper Web 2.0 tools, special visual patterns and specifications of PCA (or its specialization DCA) should be implemented during the development of

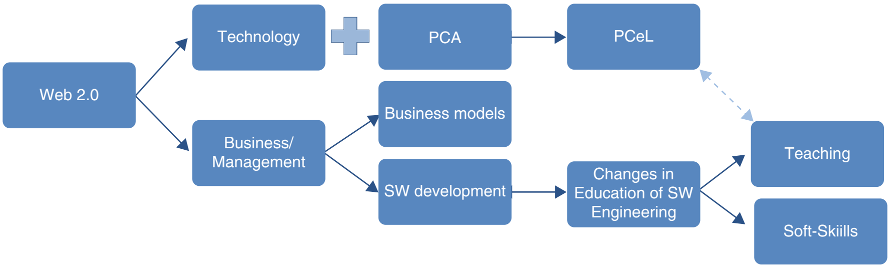


Fig. 1 Influence of Web 2.0 in technology advancement, project and business management resulting in education

software products. But generally it does not matter whether it is a desktop application, web-based services or modern mobile application in the end. The fundamental approach shall be the same.

Technically, the differences that Web 2.0 brings to ICT have important impact on project management of these services [18, 21]. Special features of modern-web based services such as development stage of "perpetual beta", absence of software release cycle, dynamic scalability, lightweight programming models, sampling and testing, etc. requires special approaches being distinguished from the classical ones [12]. Many books were written about project management and surely many will appear in the future. A basis of planning a software project, estimating its workload, building a schedule, gathering software requirements and creating use cases, improving programming with refactoring, unit testing, version control and testing software is well known. So far we dealt with approaches and software development processes which have been established and introduced into the practice (cf. [19]). These include the simple Waterfall model, Critical Chain Project Management, Extreme Project Management including Scrum and Agile Software Development to the sophisticated methodologies like PRINCE2 or Rational Unified Process by IBM. But those approaches mainly result from an earlier degree of software development appropriate for desktop programs. Web 2.0 projects have significant differences such as a rapid deployment or sharing the web-based services with many users with the ability of collaborative work accessing the data wherever the users are, while desktop programs are primarily used by one user, on one computer (or small local network) using local computer (local network) resources.

From practical experience, *Agile Project Management* is the closest approach to satisfy Web 2.0 needs so far. It is a quite new approach and therefore agile methodology reflects the current development of ICT industry tightly. Agile project management stands on simple idea of incremental or iterative (depends on what you prefer) proceedings. Thanks to this, developers are able to react on changes in time, or eventually stop the whole project, redefine it, etc.

With agile development the Scrum method is connected very closely. The basis of Scrum is so called *Product Backlog* or in other words a wish list of all things that the product shall contain. In this phase, often unrealistic requirements are included in the Product Backlog. However, immediately after the work on *Release Backlog* starts, it represents the implementation plan of particular functionalities. This may be considered as the first deviation of Web 2.0 definition from Scrum.

In Web 2.0 we often do not have very detailed implementation plans. Surely even in the Internet environment there has to be a control. But in pure understanding of Web 2.0, a new functionality is added according to the need of user, not the plan. Iteration cycle with reference to Agile methodology and Scrum is Sprint. It builds on relatively short period of time (usually 7–30 days) which satisfies Web 2.0 very short deploying cycle. Although RIA can be deployed very quickly, for instance every hour, Flickr (a famous photosharing site) has an iteration cycle a half an hour sometimes.

Therefore there is a need for project management approach combining such classical methodologies with more complex approaches reflecting the necessity of proper business models closely connected to the development, efficiency and performance of RIA also considering security issues and so on. According to contemporary trends of humanistic approaches described above, this modern approach shall not only focus on the development itself, but to developers who back up every single project as well as to users who judge the project's success. Development orientation should be also dependent on users opinion, habits and customs. Not only of control interface, but the whole software product functionality should fulfill their needs. From the very beginning, the modern IT project management shall consider this necessity.

3 User-Centered Approach in System Development

Since the information technologies have become significant part of people's everyday lives, the need to consider human perspective and various aspects of usability seems to increase.

People use technologies at work, at schools and also in their free time. Activities which depend on technologies have become more responsible (health care, process control, financial services). Moreover, not only technical-oriented people have to deal with different systems, services and devices. Current users of technologies are people with various backgrounds, motivations and skills. All these aspects indicate that orientation on usability and user needs is important.

For designing web applications, user-centered approach is particularly crucial. An application, which takes human perspective into account profits from competitive edge. If customers are satisfied, they are more likely to use new services and recommend the application to other potential users. The high number of users indicates high popularity of the product. Furthermore, if people use the product, they provide a feedback—especially, they report errors and require new features. This feedback can lead to significant improvements of services and as a result, it can enhance system as a whole. G. Donahue identified broader benefits of usability in [5]. Benefits concerning web applications are mainly reduced development, maintenance and support cost, advertising advantages and better notices in media.

3.1 Designing Usability

Designing usability introduces several requests. First of all, usability must be related to a concrete project or product. According to international standard ISO 9241 (see [6]), usability is defined as: "The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use." This means, there is no guaranteed method, which can be applied to reach satisfying results. The selection of suitable method or combination of methods depends strongly on the particular project. However, there are several recommendations which can help to develop a usable system:

- *Future users' involvement:* Future users should be involved in the process of development from the very beginning. In the stage of requirement analysis, they should be in the center of interest rather than technologies. Requirements should take into account human perspective, needs, wishes and limitations. To understand these aspects, it is important to engage users into the process. In case of designing a system that should support people while performing certain kind of activities, it is appropriate to examine the activities, ask people to explain what they like and do not like about the current way of performing activities, observe them while working, look for problems they have, and so on. Future users should also participate in later development stages—especially in system evaluation and usability testing.
  The situation is more complicated, when future users are not known explicitly. This is often the case of designing web applications—development process is not initiated by the needs of future users, it is rather started by the idea of the product owner and their business plan. In this situation, it is quite difficult to predict future users' requirements. Therefore it is suitable to ask marketing department to develop profile of target audience for the product. Characteristics of future users can be built upon the market research and design process should be oriented on representative sample of future customers.
- *Usability experts' involvement:* Usability experts should be involved in the process from the very beginning. To achieve best possible results, a usability specialist or a group of them should be devoted to deal with usability issues as a part of the project development team. They can help with specifying requirements, because they are experienced in using various techniques to identify and understand users' needs. They can suggest suitable design approach—select effective methods and techniques of usability engineering for particular product. They can also identify usability errors and find solutions to fix them. However, their work should not be focused primarily on error detection and elimination. The main reason of their participation in the project should be matching design with users needs [7].
- *Iterative process:* Development process should be iterative. Process of specifying requirements should evolve during the time—initial requirements are usually brief, they must be refined gradually and adjusted according to the situation. Iterations should be frequently discussed with future users, evaluated and redesigned. In order to make this process easier and faster, prototyping can be used. The main

reason is to identify and eliminate errors in requirement analysis, specification and design as soon as possible. Removing an error which is detected after the implementation is much more expensive and time-consuming in comparison with an early detected error.

3.2 Usability Engineering Techniques

There is a wide range of techniques that can be used to develop a usable and user-centered system [1, 4, 10]. A suitable combination for particular project can be, for instance, selected from the following activities:

- *Requirement analysis:* user and task analysis, interview, group interview, focus group, vision seminar process, survey, questionnaires, observation, study of endusers' work and context of their work,. . .
- *Prototyping:* low-fidelity, medium-fidelity, high-fidelity prototypes; simulations of activities, evaluation and redesign of prototype
- *Usability testing:* user testing, thinking aloud, cognitive walkthrough, heuristic evaluation, inspection.

3.3 Integrating Usability Activities into Development Process

Although the user-centered approach leads to many benefits (not only on the side of end-users; product owners and developers profit as well), it is usually quite difficult to enforce it in practice [11].

First of all, usability is often seen as something additional. Designing usability requires time and money. Discussion between users and developers is mediated by usability experts. It means, that communication is more time-consuming and involves more people. On the other hand, it is usually easier and more effective for usability practitioners to transform a users' expectation into requirements, because they take human perspective into account rather than technical requirements and limitations.

Another problem is, that functionality is usually preferred over usability. Benefits of usability are not tangible. Therefore, when it comes to a decision, whether to invest money and time in usability or in development of new functions, functions are often prioritized. The problem is that specified functions might not meet users' needs. The features are present in the final product but since the end-users do not need them or do not know how to use them, they are worthless. In general, it is useful trying to identify and focus on small number of functions which are interesting for most users.

Even if requirements are specified properly, there are still many ways how to fulfill them. Developers often choose the way, which is technically easy to implement. This might lead to a formal accuracy but it does not satisfy users. Very common situation is that developers claim that better solution would take much more time to implement.

Since they are responsible for technical part and other people involved in the process often do not have competence and knowledge to argue with this claim, it is usually taken as is. Therefore in this case, a project manager reduces usability requirements in order to meet the deadline and budget.

Usability testing and system evaluation is also quite expensive and time-consuming. Some companies might see this investment as additional, because users will test the product in use. This is not only inappropriate (because end-users have to cope with errors which lead to stress, inefficiency and anger) it also sheds a bad light on the reputation of product developers.

Other common problem with usability testing is that usability consultants are involved only into this late stage of development process. They are invited to perform user testing and evaluate final product. This is not convenient, because errors detected in this stage are very difficult to fix. These errors would be much easier and less expensive to eliminate in the early stages of development.

Developing a web applications introduces some additional challenges. The life-cycle of the web application is different from traditional software [2]. First of all, the process of development usually does not have clearly defined boundaries. It is rather an on-going activity of fixing errors and adding new features. Under these conditions it is difficult to determine individual stages of development process and plan usability activities suitable for particular stages.

In one sense, this might be an advantage. Iterative development process supports an opportunity to find and fix usability errors and gradually improve the application. Moreover, it also provides an opportunity to try innovative solutions for usability problems.

On the other hand, this approach can be dangerous. Usability activities are not usually part of the process. Usability is often evaluated afterwards—only after new features are implemented. It means software is not intentionally designed to be usable; it is rather created and later adjusted to be acceptable.

Potential threats of this approach are general usability reports, i.e. expert evaluations of applications provided by usability practitioners according to a set of general usability heuristics. These reports identify basic problems without taking a context of the project into account. Therefore the evaluation is not always adequate and may lead to misinterpretation.

4 Cost-Benefit Analysis of Usability

The main reason why it is difficult to enforce usability in the development process is that benefits of usable product are quite abstract. Moreover, these benefits are relative—various groups of people are involved in the process and each group has different perspective on evaluation of benefits and different priorities. Therefore it is difficult to measure and evaluate advantages and it is mostly impossible to transform all these advantages into financial gain, which is often most significant decisive factor for project managers. Main obstacle is the lack of metrics. Another problem is the need of appropriate model which allows decision-makers to evaluate investment systematically.

During the last decades, several models for justifying usability have emerged. These models use cost-benefit analysis—evaluation method for analyzing projects for investment purposes. In general, this method contains three stages:*identifying the value ofexpected cost and benefits, analyzing the relationship between these two values and making the investment decision* [3].

In terms of cost-benefits analysis of usability, a potential cost might be expressed as an amount of money spent on usability activities (salaries of employees and consultants, training costs) and benefits might be counted in connection with increased efficiency and productivity. The difference between these two values is used to demonstrate the profit which usability engineering brings to a project [7].

There are two typical examples of frameworks for cost-justifying usability [16]:

- Mayhew and Mantei's approach for cost-benefit analysis of usability activities consists of three steps: calculating costs of selected usability activities (broken down into techniques and steps); identifying and evaluating benefits relevant to a particular target audience of the analysis; estimating return on investment of incorporating selected activities into the process.
- Model designed by Clare-Marie Karat links cost-benefit analysis of usability into organizational business cases. The objective of this approach is to calculate costs and benefits of human factors work [15]. Analysis consists of three stages: identifying all expected costs and benefits of usability activities; dividing costs and benefits into groups of tangible and intangible; calculating the value of tangible benefits.

4.1 Use of Cost-Benefit Analysis of Usability

Cost-benefit analysis might provide evidence for justifying usability activities in the development process. It tries to quantify possible benefits in terms of return-on-investment, which is often significant decisive factor for project managers.

However, there are still some problems left. Cost-benefit analysis is not always compelling. It is rather a retrospective approach. It can be used to count gains of finished project, but it is more difficult to quantify profits of oncoming project. In that case estimates are necessary and might be a source of inaccuracies. There are many variables, which are estimated, these variables are multiplied and this leads to a significant bias. Moreover, these variables are interdependent and there are too many side effects, which should be taken into account (for example, when we want to estimate increase in productivity of user of the new system, we have to measure the time for performing activity; the problem is that time for performing activity differs under normal circumstances and under stress circumstances and stress might be a result of being tested).

More accurate estimations can be gained by more precise research. It means more people involved, more time spent and as a result—higher price. Other source of data might be case studies of completed projects. This is also a problem—product owners usually do not accept data of other companies as a relevant source of information. Moreover, in practice there is a lack of this kind of materials—only few companies have published a complete empirical case study with original data.

Other problem of cost-benefit analysis is that various parties have different perspectives. Different people consider different aspects as benefits. Cost-benefit analysis can hardly cover and evaluate all possible gains. One problem is diversity of benefits, other problem is the lack of metrics and models to quantify them. Most profits are rather abstract and difficult to express in terms of return-on-investment (for example reputation of company, satisfaction of employees).

All these problems cause that cost-benefit analysis is not very common tool in practice. Evaluation of cost and benefits of usability activities is usually not conducted on its own, it is rather incorporated in cost-benefit analysis of whole incoming project. However, at that time, it is too early to estimate which activities might be necessary and what benefits might be gained.

In practice, it is usually conducted by usability consultants on customers demand. It is not very often used as a tool which should enforce project managers to invest money in usability activities. Main reasons are time and financial demands, the lack of data about finished projects and high possibility of inaccuracies.

On the other hand, cost-benefit analysis is the tool which is worth knowing. It can provide project managers with useful information, because models can be used to evaluate costs and benefits of usability activities incorporated into previous projects. Based on this evaluation, it is easier to predict what activity is suitable for particular situation and what activity should be avoided.

5 Conclusion

User-centered orientation is very important nowadays, the social and humanistic aspects arose in importance. It is not an added value, but in the case of close connection with Web 2.0 it becomes an essential part of the application development itself.

The proposed cost-benefit solution, which tries to express the user orientation in terms of Return of Investment, is technically of interest to be deployed in the development process, since the project managers can easily evaluate pros coming out of the considered approaches and can thus decide about them. However, practically, it is very hard to make a correct assumption of future user needs, it incorporates long term data collection and hence is not widely used in practice. In spite of that, CBA shall be reconsidered in defining the project cycle.

The latter approach where there are investigations of past project case studies is more promising since managers have the knowledge and experience of what way may lead to better results. These new challenges with a combination of derivatives of Agile Software Development force project managers to handle both requirements from users and developers. Especially evaluation of user-centered orientation importance, proper activity planning, determining process of development, defense of this process, introducing members of development team capable to understand user needs (many programmers are not capable of doing so) and so on.

In the end, the usability of discussed principles is profitable for all parties. The users are likely to use systems which were developedon a basis of their requirements, the developers create an application which will likely be used thanks to friendly interface and managers due to higher income to their bucket.

Acknowledgements The contribution is apart of the Project (No. 2615/2010/G1) called "New Approaches to the Software Development Education" supported by the Ministry of Education, Youth and Sports, Czech Republic.

References

1. Allen B (1996) Information tasks: toward a user-centered approach to information systems. Academic Press Inc., Orlando
2. Brambilla M, Ceri S, Fraternali P, Manolescu I (2006) Process modeling in web applications. ACM Trans Soft Eng Methodol (TOSEM) 15(4):409
3. Burrill C, Ellsworth L (1980) Modern project management: foundation for quality and productivity. Burrill-Ellsworth, New York
4. Derntl M, Motschnig-Pitrik R (2004) Employing patterns for web-based, person-centered learning: concept and first experiences. In: International conference on Educational Multimedia, Hyperme dia & Telecommunications (EDMEDIA)
5. Donahue GM (2001) Usability and the bottom line. IEEE Softw 18(1):31–37
6. ISO 9241-11 (1998) Ergonomic requirements for office work with visual display terminals (VDTs)—Part 11: guidance on usability
7. Lund AM (1997) Another approach to justifying the cost of usability. Interactions 4(3):56
8. Mele D (2003) The challenge of humanistic management. J Bus Ethics 44(1):77–88
9. Motschnig-Pitrik R, Mallich K (2002) Learning at three levels: web-support and experiential learning in project management. In Proceedings of ICL 2002, International Workshop on Interactive Computer-Aided Learning
10. Norman DA, Draper SW (1986) User centered system design; new perspectives on human-computer interaction. L. Erlbaum Associates Inc., Hillsdale
11. Olson GM, Olson JS (1991) User-centered design of collaboration technology. J Organ Comput Electron Commerce 1(1):61–83
12. O'Reilly T (2007) What is web 2.0: design patterns and business models for the next generation of software. O'Reilly Media, Sebastopol
13. Paul II PJ (1991) Encyclical: Centesimus annus
14. Pitner T, Drasil P (2009) Web 2.0. DATAKON
15. Rajanen M (2003) Usability cost-benefit models-different approaches to usability benefit analysis. Citeseer
16. Rajanen M, Iivari N (2007) Usability cost-benefit analysis: how usability became a curse word? Human-computer Interaction-INTERACT 2007, pp 511–524
17. Rogers CR (1979) The foundations of the person-centered approach. Education 100(2):98–107.
18. Savidis A, Stephanidis C (2004) Unified user interface design: designing universally accessible interactions. Interact Comput 16(2):243–270
19. Stellman A, Greene J (2005) Applied software project management. O'Reilly Media, Inc., Sebastopol
20. DeTroyer OMF, Leune CJ (1998) WSDM: a user centered design method for web sites. Comput Netw ISDN Syst 30(1–7):85–94
21. Vossen G, Hagemann S (2007) Unleashing Web 2.0: from concepts to creativity. Ubiquity, (December 1)

Database-Centred Approach to WfMS and DBMS Integration

Václav Slavíček

Abstract In past, various models have been proposed to address problems posed by advanced transactions, but only a few of them are eventually used in commercial products. In this paper, we present a model which fully integrates Workflow Management System into existing Database Management System to facilitate advanced transactions. The integration can be gradual and seamless. Furthermore, we make the case that the model can be implemented by means of generally available software tools. We have designed a prototype built on Microsoft platform. Finally, we show how issues related to long-duration transactions and distributed databases can be tackled with help of the model. The experience recounted in this paper has been assembled over one elapsed year of work with Windows Workflow Foundation.

1 Introduction

Long-living transactions require distributed real-time concurrency and coordination control. Such transactions are typically shared by different participants in a loosely coupled network environment, and their execution may take a long time to complete. While long-living transactions often cannot be rolled back, they can be usually compensated. Workflow models can be understood as a superset of transaction models with added advantage of incorporating a variety of principles that have so far remained outside the scope of traditional transaction processing [2].

There has been an extensive research in the area of workflow management, database management, distributed transaction coordination as well as in integration of the formerly named areas. To cover domains ranging from distributed transaction recovery to an optimal workload balancing, scientists have suggested the use of multi-agent systems, intelligent distributed databases or even game theory. Areas like service-oriented architecture or scientific grid workflows have been affected by

V. Slavíček (✉)
Faculty of Informatics and Management, University of Hradec Kralové, Rokitanského 62,
500 03 Hradec Kralové, Czech Republic
e-mail: vaclav.slavicek@uhk.cz

J. Pokorny et al. (eds.), *Information Systems Development,*
DOI 10.1007/978-1-4419-9790-6_56,

the results of prior research. Few of these enhancements are however used in widely accepted commercial products.

In this paper, we present a model which integrates *Workflow Management System* (WfMS) into existing *Database Management System* (DBMS). The integration can be gradual and seamless, with no special tools required. Next, we design a prototype built on Microsoft Platform. Finally, some common issues related to long-living transactions and distributed database systems are tackled with help of the model.

2 Related Work

A lot of research has already been conducted in the field of database-driven workflows. Two examples are *active databases* and *advanced transaction models* (ATM). Active database enhances traditional database functionality with a rule processing mechanism, often employing event-condition-action rules [16]. ATMs represent a relaxed alternative to flat transactions.

Several techniques have been proposed to handle conflicts in long-living transactions. Examples include conflict matrices, predicatewise serializability, escrow locking, proclamation model [7], global relaxed transactions using partial compensation [5], flexible and cooperative transaction models represented by collaborative databases, and new timestamp ordering [13].

Another area of interest has grown up around complex systems with several WfMSs and DBMSs connected together in a network. These obviously bring additional challenges in Distributed Transaction Processing (DTP). There is a communication standard to support transaction-oriented cooperation between several open systems presented in [3]. Authors built their theory around the fact that a long-term business relationship involves intensity periods with information exchange interlaced by periods of quiescence. Anyway, they still name the inter-transactional conflicts as an open issue with a possible solution based on relaxed isolation.

An SQL extension to Atlas workflow is presented in [4] integrates database and workflow systems to manage scientific data. One disadvantage of the approach is that it is not fully adoptable just with generally available tools. There have been several other proposals made in the area of scientific workflows and distributed grid computation. Examples are available in [6, 14]. Nevertheless, they generally prefer parallelism (i.e. improved performance) over concurrency [12].

3 Transaction vs. Compensation

A *transaction* is defined as a basic unit of work executed on a shared system resource [13]. A standard ACID transaction possesses the following four attributes: *atomicity*, *consistency*, *isolation* and *durability*. These properties predetermine the time span between starting and committing a transaction to be very short. On the other hand, long-living transactions support from minutes to days long duration at the expense of impaired atomicity and isolation. However, the consistency and durability properties remain unaffected. Long-living transactions usually contain either standard atomic transactions or alternatively other long-living transactions nested to arbitrary depths.

When cancelling a long-living transaction, an operation called *compensation* is used rather than roll-back. This is due to the lack of isolation, because partial effects of the incomplete transaction have been visible to all participants in the interim. Unlike reversing in time with no impact at all when performing a roll-back, compensation is more a sort of message. The message announces a cancellation of the process which has already been published.

Theory of Distributed Transaction Processing (DTP) classifies data manipulation inside and outside a transaction scope as *protected* one and *real* one, respectively. While *real* actions cannot be "undone", they can be compensated [9]. DTP is an essential feature of Distributed Database Systems (DDS). DDS is a collection of multiple, logically interrelated databases distributed over a network [11]. Reasons for the physical decentralization can range from an improved performance (when reflecting organizational structure, for example) to economic or even geopolitical aspects.

4 Transaction Processing in Workflow Management Systems

The experience of past two decades proved that, despite of indisputable advantages, relational databases suffer from numerous weaknesses such as poor representation of real world entities and homogenous data structure. Object-relational DBMSs or OR-Mappers are only two alternatives to overcome these issues. In a similar way, *Workflow Management System* (WfMS), which typically provide an IT infrastructure for robust enactment of both human and machine-driven business processes, can be also apprehended as a superstructure over a standard DBMS, when managing control-flow similar to how DBMS handles data-flow. Its main advantage comes with the ability to change the state of particular workflow instance, based on incoming events.

A typical architecture of WfMS integrated with DBMS is depicted in Fig. 1. It is obvious that changes in control-flow require some synchronization with the changes made to the data. Let's take an example where a bank clerk approves a credit claim. Workflow instance within the IT infrastructure changes its state from *BeingEvaluated* to *Approved.* At the same time, the appropriate data object must be modified, so that details of the claim will be stored in a *Case* DB table which holds the claim-related data. To sum up, both control-flow and data-flow should occur in the same distributed transaction, often spanned over the network by means of persistent message queues [8].

Workflow instance is usually identified by a *unique identifier* (UID). This UID identifies particular instances in WfMS and so it does in the database where these instances persist. The same UID also refers to the database with table *Case* which stores the instance-related data. In context of DDS, there will be a vertical fragmentation with one fragment residing on WfMS (in the persistence store) and other

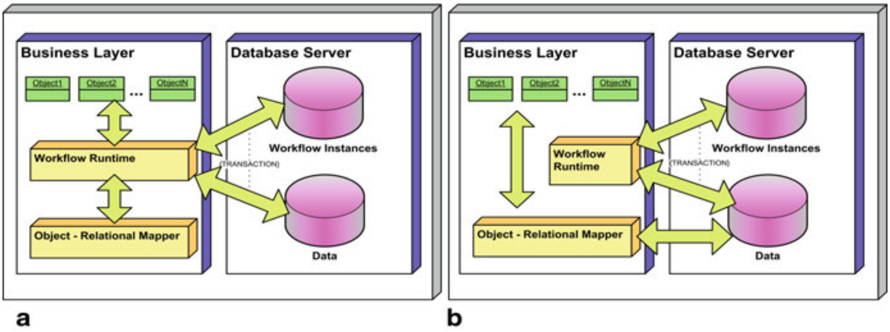


Fig. 1 **a** Typical architecture. WfMS resides it the centre of communication. **b** Database-centred model extends the functionality of standard DBMS

fragment in DBMS:

$$S_{\text{WfMS}} : \prod\nolimits_{\text{UID, Status,...}} (\text{Case}) \tag{1}$$

$$S_{\text{DBMS}} : \prod\nolimits_{\text{UID,Claim-related data}} (\text{Case}) \tag{2}$$

It would not be a rare situation in which several workflow models share a common WfMS yet different DBMSs. In such case, we can talk about a derived fragmentation. Another remarkable phenomenon relates to workflow implementation. Workflow instances often contain internal structures that hold data about the object, identical to those stored in DBMS. The only reasons for such redundancy are an easier access to the data within WfMS, and better performance. Then we can talk about data replication across heterogeneous distributed database.

5 Database-Centred Model of Integration

In a word, existing work commonly suffers from the following drawback: The suggested approaches are not applicable using most widely used WfMSs and DBMSs. In contrast, our model only requires features that are present in solutions provided by all major software vendors. The requirement of straightforward implementation is deemed essential for every model to advance from academia to industry. In contrary to the standard approach in Fig. 1a, b introduces model suggested in this paper. This time, business layer communicates with WfMS, in the picture referred to as *Workflow Runtime*, solely via DBMS referred to as database *Data*. WfMS in turn persists idle workflow instances in the persistence store referred to as database *Workflow Instances*. We use a DB trigger to notify WfMS about changes performed on data. This requires DBMS to provide some basic interoperability. Both Microsoft (since MS SQL 2005) and Oracle (since Oracle9i) offer a functionality to invoke an external code written in .NET, or Java, respectively [10].

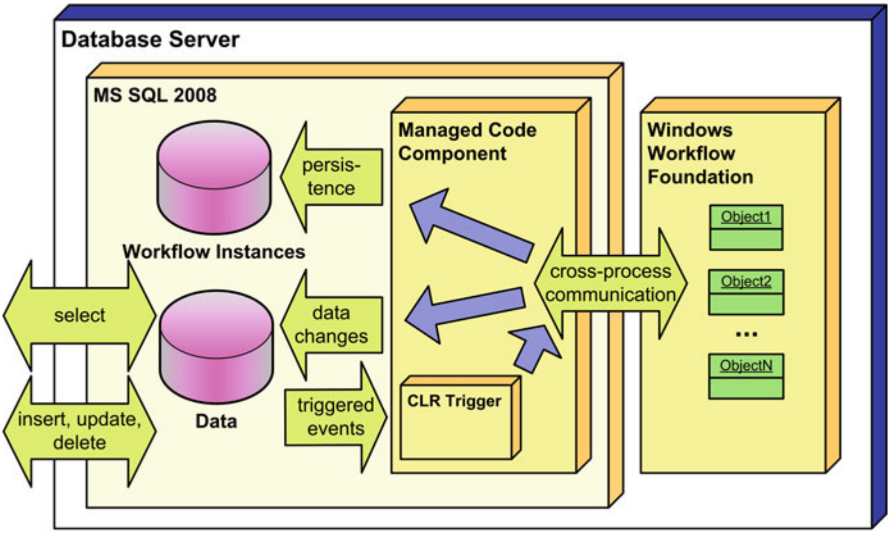


Fig. 2 Detailed communication in the database-cantered model

5.1 Implementation of Prototype

We have decided to build a prototype in order to prove the principles stated above. The prototype has been implemented on the Microsoft platform, using MS SQL 2008 and .NET 3.5. Windows Workflow Foundation (WWF) is used to design and enact workflow programs. The out-of-box *SqlPersistenceService* has been slightly modified to serve as a persistence store [15]. Figure 2 contains details of architecture and communication among particular components.

We use a *CLR* "instead of" *trigger* to notify WfMS about a data-related event. The trigger has been written in C#, and compiled for .NET Common Language Runtime. Theoretically, whole WWF could be fully loaded within the MS SQL Server process (*sqlservr.exe*), although it is not recommended to load unproved libraries directly to the server. Better alternative is to execute the workflow runtime separately as a Windows service and to use some form of cross-process communication. Hence we have decided to isolate DBMS and WfMS, and to traverse the processes by means of .NET Remoting. Scheduled pending instances are executed in the synchronous model [1]. *Delay activities*, otherwise asynchronous by their nature, are executed with the help of *active timer*. This in turn facilitates implementation of advanced compensation/data replication scenarios.

5.2 Example

To illustrate how the proposed model works, let's consider a system for automated credit claim evaluation. Each case of claim is represented by a record within DB table

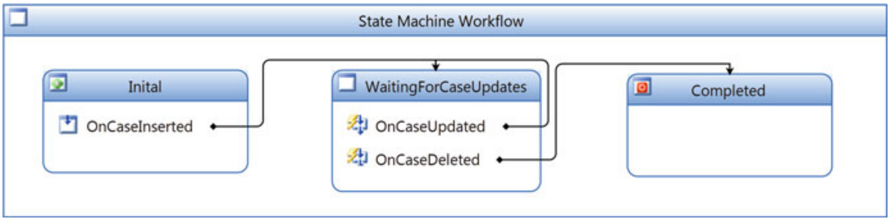


Fig. 3 This state-machine workflow provides the same functionality as a standard DBMS

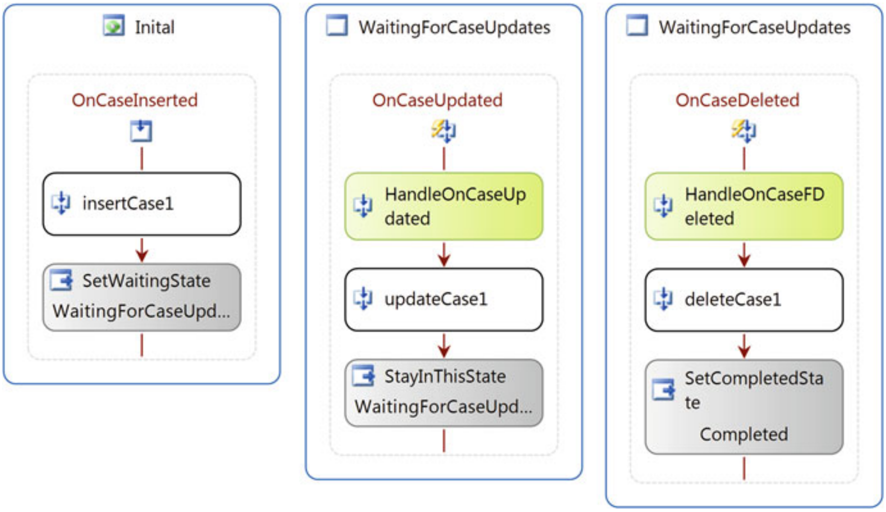


Fig. 4 Sequences that are triggered when specific events fire for the database table Case. They assure that the required operation will be performed for the appropriate row. Finally, state of the workflow is changed appropriately

Case. The table contains two columns: *CaseId* uniquely identifying each claim in the table, and *StatusId* holding one of the following states: *Initialized, BeingEvaluated, Approved,* and *Rejected*. It is an ordinary DB table so far. Our model, however, also implements the following workflow-related infrastructure behind the table:

- Predefined workflow activities *InsertCase*, *UpdateCase,* and *DeleteCase*. The activities require input describing the row, and perform specific operation against data in table Case. *InsertCase* inserts new row. Analogically, *UpdateCase* updates, and *DeleteCase* removes specified row from the table.
- CLR "Instead of" trigger is designed to handle insert, update and delete events when such an operation requested by an SQL command against the table.
- State-machine workflow (see Fig. 3) which by default ensures that the DBMS will behave exactly the same way as a standard DBMS. Figure 4 reveals particular sequences for state transitions. These sequences are triggered once an appropriate event (*insert*, *update*, or *delete*) fires for the table.

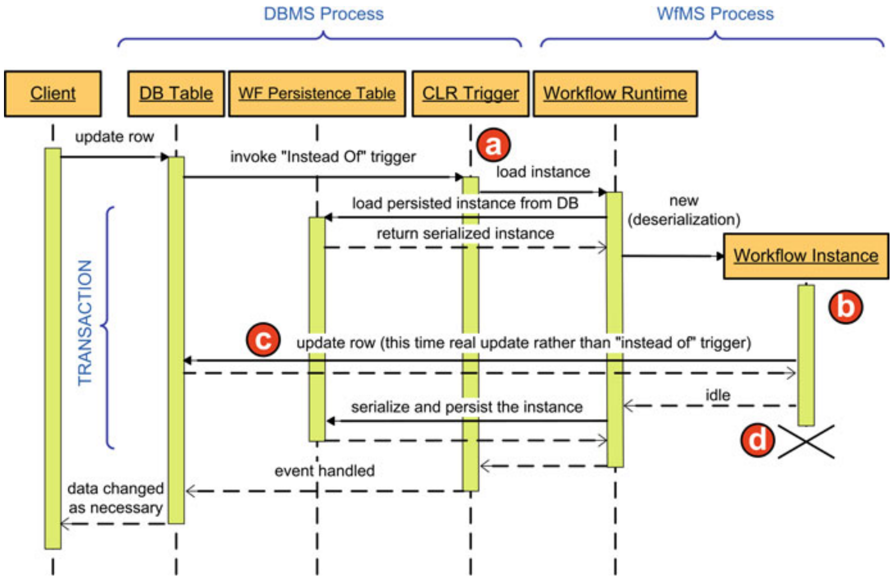


Fig. 5 Diagram representing communication behind a single DB update

Let's take a look at a typical scenario, such as an update of the *Case.StatusId* column. Details of the background communication are captured in Fig. 5.

1. SQL statement *UPDATE TABLE* invoked against DBMS normally leads to an update within the scope of implicit transaction. The "instead of" trigger causes a difference, though. With this trigger, no data are modified directly in the table. Instead, DBMS invokes WfMS via a cross-process channel. WfMS first restores the appropriate workflow instance which state has been previously persisted. We presume that the instance has resided in the *WaitingForCaseUpdates* state.
2. Event *CaseUpdated* is delivered to the instance, and the instance executes what is planned in the workflow model.
3. During this execution, the instance invokes the *updateCase1* activity. This actually updates the table the same way as it would be normally done in step *a*.
4. Finally, the updated instance state gets saved back to the persistence store.

This approach lets us share a single transaction within the whole trigger execution, handing it seamlessly from DBMS (i.e. SQL Server) over to the WfMS (i.e. WWF in .NET). Furthermore, WWF provides a built-in batching mechanism as well as a support for *2 Phase Commit*.

5.3 *Model Utilization*

The proposed model brings several specifics we can take advantage of. To target security, for example, it is possible to revoke direct access to a specific column,

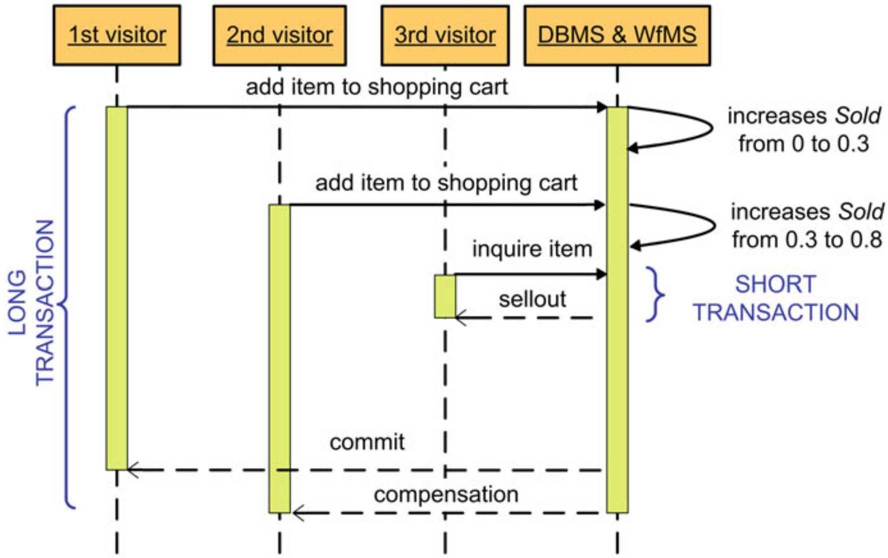


Fig. 6 Evaluation of several concurrent participants within a single long-living transaction. Example of reasoning for tree potential buyers. The third buyer is rejected immediately, since the item is considered unlikely to stay in stock

so that a DB user can only modify it via workflow engine. Another convenient by-product is an audit trail. WWF offers out-of-the box *tracking service*. This service logs events of specified type to the database. These logs can facilitate a process mining, an implementation of sophisticated partial roll-backs, transaction compensations, etc.

The following two subsections introduce in more detail how the model can help with long-living transaction handling, and with DB replication.

5.3.1 Long-Living Transactions

In long-living transaction management, the philosophy usually consists in sacrificing isolation with the view of better concurrency. Concurrency control is a process of managing simultaneous operations without having them interfere with one another. Concurrency-related issues can be further classified as uncommitted dependency, lost update, and inconsistent analysis [4]. There are two basic types of concurrency control algorithms established in the database theory. Pessimistic algorithm prevents the concurrent transactions from accessing the data since the beginning, whereas the optimistic locking solves a potential conflict by a transaction abort ex post. The first approach is only suitable for operations with short duration, while the latter one reveals uncommitted data [7]. Advanced Transaction Models (ATM) can benefit complex solutions with long-living transactions and no prior knowledge of where the model further evolves. Detection and management of conflicts comes in several specific forms, ranging from compensation-based Sagas to nested transaction levels.

In the following example, we will adopt the approach of Multilevel Transaction Model. This model introduces a hierarchy of transactions where those at higher level may not conflict even though their implementations at lower levels do conflict. [4]. Let's consider an optimistic lock used during a long-living transaction in an online shop, as depicted in Fig. 6. If several potential buyers have added an identical item to their shopping carts concurrently, still only one can make the deal. This situation often happens for commodity which is scarce. An example is flight tickets.

Using a typical approach, an item in the shop would be considered strictly either sold or not sold. A pessimistic algorithm would consider the item sold at the beginning of transaction, whereas an optimistic one at the commit. It is obvious that although shopping activities of two concurrent buyers do conflict, there is no conflict in general, until the business is fixed with the quickest buyer.

For our purpose, rather than using crisp *Boolean*, we will represent property *Sold* by *Real* ranging from 0 to 1 to represent the probability of sale. Various precursors can be used to estimate this value. Examples are the number of visitors interested in the item, their previous shopping history etc.

The suggested approach helps us to balance concurrency needs with overall throughput, so that both quality of service and quality of data remain acceptable.

5.3.2 Selective Asynchronous Replication

Data updates in a distributed environment can be either performed immediately, typically by means of a 2PC (Two-phase commit) protocol, or in more "relaxed" way later on. The letter approach is called *asynchronous replication* [4]. The following example is confronted with the challenge of intelligent redistribution of data changes after an update performed against a single site in a distributed database system involving a single client and a distributed DBMS with two sites. Although today's DBMSs provide tools for replication, the model presented in this paper can bring even more sophisticated yet straightforward methods.

There are two alternatives of asynchronous replication depicted in Fig. 7. The alternative on left updates Site 1 immediately as part of the enclosing transaction. Considering the model introduced earlier, workflow instance invoked by the update event may becomes idle again, waiting to be activated internally several seconds or even days later. An active timer can be scheduled to reactivate the workflow in order to update Site 2 at a specific time, on a specific request (E.g. a request for summary report at an analysis stage of data warehouse), or at the time of low network traffic.

Figure 7b represents another alternative. Here, the request to update the sites is part of an enclosing transaction, but in an asynchronous way, i.e. without waiting for the response from Site 2. We presume that the update of Site 2 is time consuming. When finished, Site 2 reactivates the workflow instance on Site 1, and commits the long-living transaction, eventually. Should it happen the asynchronous update fails, there are various ways how to perform compensation.

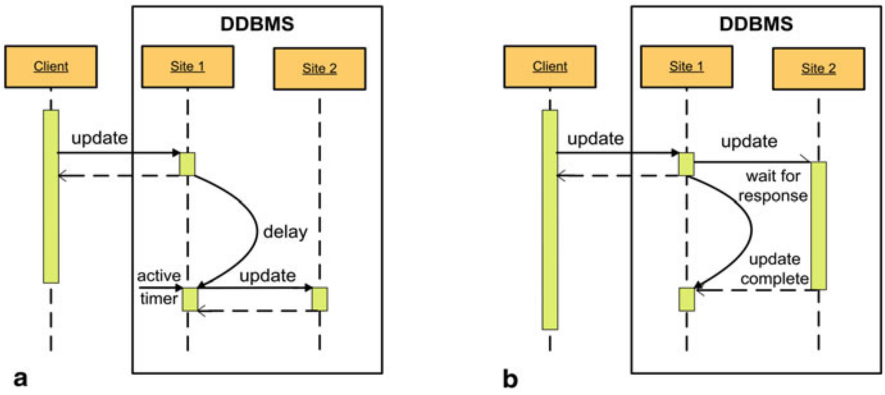


Fig. 7 Two alternatives of asynchronous replication: **a** postponed update, **b** asynchronous update

There is another advantage resulting from technical implementation of the model: WWF provides ready-to-use activities designed to work with web services. These can facilitate remote communication within heterogeneous DDBMS.

6 Conclusion and Future Work

In this paper, we introduced a model seamlessly integrating DBMS with WfMS. It can bring improvements to DBMSs by the use of workflows triggered directly by SQL commands. As proved on the designed prototype, the workflow functionality is easy to implement, maintain, and monitor. When testing the model, we focused on implementation of sophisticated techniques to handle long-living and/or distributed transactions.

We proved the model worked and has brought numerous advantages. One of them is fragmentation transparency when used in distributed environments. Yet another advantage resides in use of an "instead of" trigger for DB inserts to initialize associated processes even during mass imports. On the other hand, there comes a substantial drawback with "instead of" triggers since they are poorly transparent by nature. Hence a less experienced DB user can face difficulties when trying to understand what is actually happening on behind.

There are other interesting principles which would be worth to test against the proposed model. One of them is workflow replication. In this concept, an item is passed across distributed databases depending on its state. An invoice, for instance, can travel from site to site in order to be lined up, approved, and finally paid. Other possibilities include a search for optimal data distribution pattern in DDBMS, as well as an evaluation of the model against database-centred methods of business intelligence, such as data warehousing, OLAP, or data mining.

Acknowledgements This work was partially supported by the Czech Science Foundation project No. P403/10/1310 SMEW as well as by the FIM UHK internal project IPSV1/10 AmIRRA.

References

1. Allen KS (2006) Programming Windows workflow foundation. Packt, Birmingham. ISBN:1-9048-1121-3
2. Alonso G et al (1996) Advanced transaction models in workflow contexts. In: Proceeding of 12th international conference on data engineering. IEEE Computer Society, Washington, pp 574–581. ISBN:0-8186-7240-4
3. Banks R et al (1998) OSI distributed transaction processing commitment optimizations. ACM SIGCOMM Comput Commun Rev 28(5):61–75. ACM, New York, Oct. ISSN:0146-4833
4. Connoly TM, Begg CE (2009) Database systems: a practical approach to design, implementation and management. Addison-Wesley, Harlow. ISBN:0-321-52306-7
5. Grefen P et al (1999) Semantics and architecture of global transaction support in workflow environments. In: Proceedings of the 4th IECIS international conference on cooperative information systems. IEEE Computer Society, Washington, p 348. ISBN:0-7695-0384-5
6. Jianwu W et al (2008) A high-level distributed execution framework for scientific workflows. In: Proceedings of the 2008 4th IEEE international conference on eScience. IEEE Computer Society, Washington, pp 634–639. ISBN:978-0-7695-3535-7
7. Kuo D, Gaede V, Taylor K (1998) Using constrains to manage long duration transactions in spatial information systems. In: Proceedings of the 3rd IFCIS international conference on cooperative information systems. IEEE Computer Society, Washington, p 384. ISBN:0-8186-8380-5
8. Leymann F, Roller D (1998) Building a robust workflow management system with persistent queues and stored procedures. In: Proceedings of the 14th international conference on data engineering. IEEE Computer Society, Washington, pp 254–258. ISBN:0-8186-8289-2
9. Lindsay B (2008) Jim Gray at IBM: the transaction processing revolution. ACM SIGMOD Rec 37(2):38–40. Special issue: Proceedings of tribute to honor Jim Gray, June 2008. ACM, New York. ISSN:0163-5808
10. O'Rourke C (2003) The active database: ways to invoke external processing from the database. Oracle Magazine, Sep 2003. Oracle, Redwood. ISSN:1065-3171
11. Özsu MT, Valduriez P (1999) Principles of distributed database systems. Pearson Education, Harlow. ISBN:8-1775-8177-5
12. Qin J, Fahringer T (2007) Advanced data flow support for scientific grid workflow applications. In: Proceedings of the 2007 ACM/IEEE conference on supercomputing. Art nr. 42. ACM, New York. ISBN:978-1-59593-764-3
13. Ramampiaro H, Nygård M (1999) Cooperative database system: a constructive review of cooperative transaction models. In: Proceedings of the 1999 international symposium on database applications in non-traditional environments. IEEE Computer Society, Washington, p 315. ISBN:0-7695-0496-5
14. Shankar S et al (2005) Integrating databases and workflow systems. ACM SIGMOD Rec 34(3):5–11. ACM, New York. ISSN:0163-5808
15. Shukla D, Schmidt B (2006) Essential Windows workflow foundation. Addison-Wesley, Harlow. ISBN:0-3213-9983-8
16. Widom J, Ceri S, Dayal U (1996) Active database systems: triggers and rules for advanced database processing. Morgan Kaufmann, San Francisco. ISBN:1-55860-304-2

The Role of Implementation Strategy in Enterprise System Adoption

Piotr Soja

Abstract The goal a of this study is to investigate issues connected with implementation strategy in enterprise system (ES) adoption. Drawing from the experience of a few dozen ES adopters, this study examines how different implementation strategies exist in practice, what project characteristics are connected with the choice of an implementation strategy, and how implementation strategy influences ES adoption outcomes. In doing so, this study employs various success metrics which include three-dimensional project success measure and user satisfaction. The results demonstrate that project duration time is the main factor deciding about the implementation strategy adopted. They also suggest that various risks are connected with implementation strategy choices. In particular, phased method is connected with longer adoption time and the risk of exceeding the planned time, while parallel method runs the risk of exceeding the planned financial budget. On the basis of the research, recommendations regarding the implementation strategy choice conclude the study.

1 Introduction

Enterprise systems (ES) are complex software application packages that promise the seamless integration of all the information flowing through a company—financial and accounting information, human resource information, supply chain information, customer information (Davenport 1998). ES adoption is usually connected with a multi-phased and complicated implementation process which is caused by the system complexity and organizational context. In general, it is virtually impossible to implement an ES with no organizational change (Newell et al. 2005). In fact, during ES adoption the company has to work out the solutions in order to get rid of both organisational and technological misalignment between the organisation and ES system (Ho et al. 2004).

P. Soja (✉)
Department of Computer Science, Cracow University of Economics, Rakowicka 27, 31-510 Kraków, Poland
e-mail: eisoja@cyf-kr.edu.pl

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_57,

ES adoption can take various shapes depending on particular company and project conditions. ES implementation projects range from a relatively simple introduction of an ES in a single plant to complex multi-national implementation projects covering many branches and requiring many changes in organisation structure (Parr and Shanks 2000b). There are a great many issues that may affect ES adoption, they are often expressed in terms of critical success factors (CSFs). Critical issues that may affect ES adoption include implementation strategy, which is considered as one of strategic CSFs for ES adoption (Holland and Light 1999). Further, Mabert et al. (2003) claim that the strategy used for the implementation is one of the most important factors in determining the outcome of an ES project.

The purpose of this research is to explore how different implementation strategies exist in practice, and to discover why companies decide on using a particular approach. The actual research question involved in this study can be formulated as follows: *What implementation strategies do companies adopt and how is it connected with different project's characteristics and outcomes?*

2 Enterprise System Adoption Process

2.1 Adoption Stages

ES adoption is usually a multi-stage process during which the project preparation and planning take place and are followed by the system installation and use. There are various approaches and propositions of implementation project stages which divide the whole process into consecutive phases. The most important propositions from the extant literature contain from three to six stages and are shortly described below.

Parr and Shanks (2000a) divide implementation process into three general phases: Planning, Project, and Enhancement. Within the Project phase, they distinguish five sub-phases: Set up, Reengineering, Design, Configuration and testing, and Installation. The general categorisation of an ES adoption process into three stages is also present in the model of Al-Mashari et al. (2003), who recognise the following phases of an ES project: setting-up, deployment, and evaluation. Finally, Deloitte Consulting (1998) claim that there are three distinct stages after going live: Stabilize (secure and sustain the core ERP functionality), Synthesize (build for the future by adding still other capabilities—often non-ERP—to the mix), and Synergize (achieve value in use by thoroughly mastering those new capabilities).

Markus and Tanis (2000) discern four main enterprise system adoption phases named: Project chartering, The project, Shakedown, and Onward and upward. During the Project chartering phase the company makes decisions defining the business case and solution constrains. The project stage involves system configuration and rollout with the purpose of getting system and user "up and running". During the Shakedown phase the organisation is stabilising, eliminating "bugs", and getting to normal operations. Finally, the Onward and upward stage includes maintaining system, supporting users, getting results, and upgrading.

Ross and Vitale (2000) suggest five adoption stages and in doing so, they humorously claim that the phases of an ES implementation resemble the journey of a prisoner escaping from an island prison. The proposed ES adoption stages are: design (the approach, in the case of a prisoner), implementation (the dive), stabilisation (resurfacing), continuous improvement (swimming), and transformation (from prisoner to free man). The authors interestingly discuss the company's performance over the stages of ES adoption project and emphasise the presence of a performance dip during the implementation stage.

Somers and Nelson (2004) distinguish six implementation phases and base their proposition on the Rajagopal (2002) framework of ERP implementation in terms of six-stage model of IT implementation. The proposed stages of ES implementation include: initiation, adoption, adaptation, acceptance, routinization, and infusion. The authors claim that the strength of this model is in the last two phases, which represent post-adoption behaviour.

Finally, apart from the models proposed by the researchers, it is valuable to present the implementation approach named Accelerated SAP (ASAP) worked out by the global leader on ES market: SAP. The ASAP framework divides an implementation project into five phases (e.g. Lui and Chan 2005):

- Project Preparation—initial planning and preparation for an SAP implementation project, defining a unique objective, project scope, priorities, etc.
- Business Blueprint—with the purpose to create a detailed document describing how the company is running its business before implementing the SAP applications ("As-Is" model) and how it intends to do so afterwards ("To-Be" model).
- Realization—implementing business and processing requirements based on the blueprint, establishing the required "To-Be" environment.
- Final Preparation—finalizes everything in readiness for going live with the new systems and processes; involves testing, end-user training, system management, and "cut over" activities.
- Go Live and Support—with the purpose of moving a pre-production environment to live production operation; includes setting-up a help desk and long-term support for end-users, monitoring system transactions and optimizing overall system performance.

2.2 Implementation Strategies

All implementation methods, regardless of the number of stages, their scope etc., have to include an event consisting in the launching of the new system and moving the company to operations supported only by the new system. During this event, legacy systems operating previously in the company are abandoned. The ultimate effect of this process is the operation of an ES package on a daily basis within the whole company.

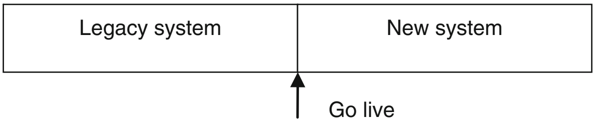


Fig. 1 Big-bang implementation strategy

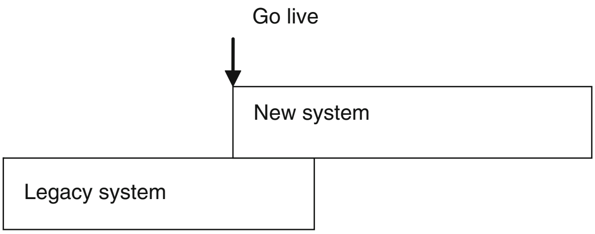


Fig. 2 Parallel implementation strategy

There are different implementation strategies used in this process and the main division consists in the distinction between a big-bang and phased approach (e.g. Holland and Light 1999). The big-bang approach is also called "total cutover" or "cold turkey" (Umble et al. 2003). In this implementation strategy, the whole company is eventually brought onto the new system with its complete functionality in a single effort (see Fig. 1). The entire company prepares for the cutover date, which would preferably be during a plant shutdown of 1–2 weeks. The big-bang strategy is considered to be highly risky (e.g. Markus et al. 2000).

The opposite implementation approach is called phased, gradual, incremental or progressive (e.g. El Amrani et al. 2006; Robey et al. 2002). The main idea here is that the new system is introduced gradually and it is used together with legacy systems for a certain period of time. In a phased approach, the new system functionality is introduced into the company sequentially. This can be done by modules, products or plants. After the first module/product/plant is introduced, procedures may be refined and adjusted, then the remaining modules/products/plants are sequentially implemented (Umble et al. 2003). The main advantage of this approach is that it allows for improvements to be made during the implementation.

Nonetheless, the distinction between a phased/gradual and a big-bang implementation is too coarse to capture the range of actual approaches to ES implementation which are existing in industry (Parr and Shanks 2000b). In particular, a parallel strategy can be distinguished, which can be treated as a border approach within the range of gradual methods. In this approach, there is a period of time during which legacy systems and the new system are used simultaneously within their full functionalities (see Fig. 2). This method requires extraordinary effort from employees since each transaction must be entered into the legacy system and then into the new system (Okrent and Vokurka 2004). Decreasing gradually the legacy system functionality used in the organization, we approach to the "classical" phased approach, as illustrated in Fig. 3. As noted above, this phased approach can be performed by modules, products or plants.

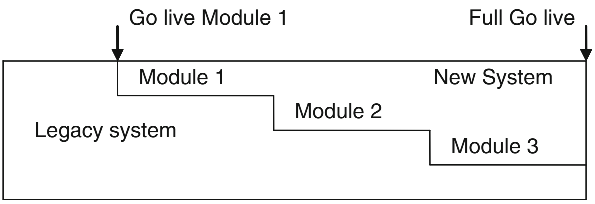


Fig. 3 Phased implementation strategy

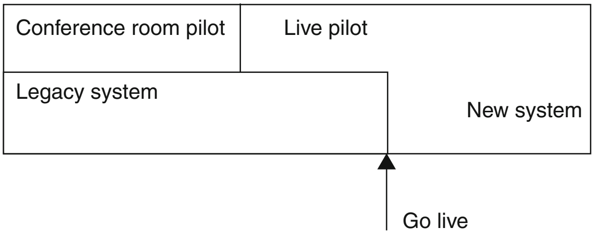


Fig. 4 Pilot implementation strategy

The next type of a gradual approach to ES implementation is called pilot implementation. In this approach, a specific functional area is implemented first. The idea is to prioritize the functional areas and implement them in a specific order that provides the most benefit first (Okrent and Vokurka 2004). The underlying principle is that the new system is used within a subset of the organisation and verified over a period of time. Within this approach, two version of pilot steps are distinguished: Conference room pilot and Live pilot (see Fig. 4). The conference room pilot exercises the system and tests the users' understanding of the system (Umble et al. 2003). Consequently, during Live pilot the new system is being operated with the use of actual data, together with legacy systems. The goal of this step is to prove that the ES package is working correctly and is accepted by users (Soja 2007). The pilot approach is considered the lowest risk alternative (Okrent and Vokurka 2004).

3 Research Methodology

In order to answer the research question, this study builds on exploratory research conducted among companies adopting ES in Poland. A field study was used as a general research approach and a questionnaire was employed as a data-gathering technique (Boudreau et al. 2001). The purpose of the questionnaire was to gather data connected with implementation strategy used and the project characteristics and outcomes. The research questionnaire was directed toward people playing leading roles in the adoption.

The respondents were asked about the implementation strategy used and they were presented the list of predefined approaches. This list included the following methods: big-bang, parallel, pilot, and phased by products or modules. The respondents were also able to provide comments in an open form.

Table 1 Companies and projects characteristics

Number of employees	%	Module	%	Duration time (months)	% planned	% actual
0–50	4.4	Finance	95.6	0–6	17.6	13.2
51–100	4.4	Inventory	92.6	6–12	29.4	29.4
101–200	16.2	Sales	86.8	13–18	27.9	20.6
201–300	17.6	Purchasing	85.3	19–24	5.9	13.2
301–500	17.6	Shop floor control	60.3	25–36	13.2	10.3
501–1000	20.6	MRP explosion	47.1	Above 36	4.4	10.3
Over 1000	19.1					

Table 2 Implementation strategies and project characteristics

Implementation strategy		%	Avg. # of modules	Avg. # of employees	Planned months*	Actual months*	Relative time
Overall		100.0	4.68	828	12.6	15.6	1.29
Big-Bang		36.8	4.80	591	8.2	8.5	1.17
Gradual		63.2	4.60	965	15.0	19.6	1.35
	Phased	33.8	4.87	751	17.1	23.6	1.51
	Pilot	14.7	4.30	955	9.4	11.9	1.11
	Parallel	14.7	4.30	1469	15.8	17.5	1.20

Note: *$p < 0.03$ as indicated by Kruskal-Wallis test

The researched project characteristics included company size in terms of the number of employees, adoption scope defined by the number of introduced modules, and planned and actual ES adoption duration time. The investigated ES adoption outcomes involved success metrics defined as end-user satisfaction (e.g. Somers et al. 2003) and project success in terms of three dimensions measuring relative duration time, budget, and implementation scope (e.g. Markus and Tanis 2000).

As a result, 68 opinions were gathered from various organisations regarding their ES adoption experience. The researched projects implemented various ES packages, both national and globally recognised, with SAP (25%) and IFS Applications (10%) as the most popular system solutions. 75% of investigated organisations were manufacturing enterprises. Table 1 presents the investigated projects characteristics. It shows, in consecutive columns, the percentage of companies/projects depending on the number of employees, the most important system modules implemented, and planned and actual project duration time.

4 Results

Table 2 contains data describing implementation strategies used by the companies and related characteristics of the companies and projects. The table includes the average values of the following characteristics: number of implemented modules, number of employees, planned and actual project duration time in months, and

Table 3 Implementation strategies and success metrics

Implementation strategy		Project success				User Satisfaction
		Overall	Time	Budget	Scope	
Overall		0.80	0.70	0.83	0.86	0.66
Big-Bang		0.82	0.74	0.84	0.87	0.63
Gradual		0.79	0.67	0.83	0.86	0.67
	Phased	0.75	0.61	0.81	0.84	0.70
	Pilot	0.87	0.76	0.94	0.90	0.64
	Parallel	0.78	0.73	0.76	0.85	0.64

relative measure of time expressed by the ratio of actual to planned adoption time. Data indicate that 37% of investigated projects employed the big-bang strategy. The remaining projects used various methods phased in time, collectively named here as gradual approaches. Within this group of adoptions, phased method was the most popular (34%), employed mainly by modules and then by products. The pilot and parallel methods were less popular, each used by about 15% of companies.

During the analysis, statistical techniques were used to find out if significant differences exist between extracted groups of projects. Namely, non-parametric Kruskal-Wallis test was adopted to assess the significance of the mean values of project parameters and success measures.

Upon analysing the parameters of ES projects, it can be noted that, as far as implementation scope is concerned, there are not big differences among implementation strategies. There is a slight tendency among pilot and parallel approaches to adopt less modules, and phased methods implementing the largest number of system modules. On average, the big-bang approach was selected by the smallest companies, and the gradual method was employed by larger firms.

Project duration is the characteristic having the strongest connection with implementation strategy. The data show that the shortest projects opted for the big-bang method, while the longest implementations employed the phased and parallel methods. It is worth noting that there is a statistically significant difference among mean values and this applies both to planned and actual duration times. Further, in order to illustrate how implementation strategy influenced the lengthening of implementation time, a ratio of actual to planned adoption time was calculated and its average values were presented in a column “Relative time”. The data indicate that the planned implementation time was clearly exceeded in the case of phased approach.

It is worth mentioning that the respondents sometimes indicated the combination of several implementation methods and most often indicated one leading method. In these cases, the projects were classified on the basis of the leading method. It turns out that the phased method was most frequently indicated and was combined with other phased, parallel or pilot approaches.

Table 3 includes average mean values of success metrics depending on implementation strategy. There are presented partial measures of project success concerning time, budget and scope, together with the overall project success measure. A separate column presents user satisfaction, which can be treated as an overarching measure of a long term ES adoption success (e.g. Sedera and Tan 2005).

The data regarding success measures indicate that project success is the biggest in the case of the pilot method and is the smallest among projects adopting the phased approach. As far as user satisfaction in concerned, there is a reverse relationship in comparison with project success. Namely, the phased approach is clearly connected with greater user satisfaction compared to the remaining methods, which, including big-bang approach, achieved considerably worse results.

5 Discussion of Findings

Upon analysing this study's results, two issues are worth noting, expressed also in the research question: how implementation strategies chosen by companies are connected with company and implementation project parameters, and how implementation approach influences adoption success. The first problem touches upon the general problem of fit between implementation strategy and a company and its intentions.

This study's findings show that planned implementation time is a project's characteristic which first and foremost influences implementation strategy choice. It turns out that the big-bang approach is significantly connected with shorter adoption type compared to the gradual approach. In the gradual method, implementation time is significantly longer, nonetheless, it is worth noting that we have some diversity here. It can be noticed that within gradual approaches, implementation projects using the pilot method are considerably shorter than phased and parallel approaches.

Findings regarding the significant influence of adoption time on the chosen implementation strategy are confirmed by prior research. Mabert et al. (2000), on the basis of a survey conducted among American manufacturing companies, state that firms able to employ a big bang approach experienced the shortest implementation time and conclude that, in general, a phased implementation increased time. Further, the study by Olhager and Selldin (2003) conducted among Swedish manufacturing firms reveals that the average implementation time is shorter for companies employing a big bang approach to ES adoption.

The second issue influencing implementation strategy relates to company size. The results suggest that smaller firms reveal a tendency to choose the big-bang implementation approach. This is in accordance with prior research which illustrate that large companies tend to use the phased implementation strategy in their ES systems and, in contrast, the most common strategy followed by smaller companies is the big-bang approach (Duplaga and Astani 2003).

Considering the influence of implementation strategy choice on ES adoption successfulness, we can distinguish an immediate project success illustrating mainly the quality of project management, and more far-reaching adoption success measured by user satisfaction. In general grouping of implementation approaches into big-bang and gradual, the results suggest that big-bang projects cope somewhat better. Simultaneously, there is an interesting situation among gradual methods, where it can be seen that projects using the phased approach are in the best condition, while pilot and parallel projects are worse off.

The examination of partial measures suggests that the poor performance of phased projects are caused first and foremost by the partial component related to time, which is definitely the lowest for this implementation strategy. Therefore, phased projects not only last longer than other implementations according to plan, but also are especially subject to exceed the planned duration. The analysis of another partial measure connected with budget reveals that parallel projects are connected with the greatest risk of overloading a company's financial budget. This is due to the fact that parallel implementations have the lowest budget measure.

The leading role of projects employing pilot or big-bang approaches significantly changes when we take into consideration user satisfaction measure. In turns out that projects of these types reveal the smallest values of user satisfaction. In contrast, the highest value of user satisfaction is connected with the phased approach. Hence, it turns out that an extended implementation time may have a positive influence in the form of better ultimate outcome measured by user satisfaction. On the other hand, as suggested by the data, keeping project assumptions may be connected with lower user satisfaction, which can be noticed in the case of the big-bang and pilot methods.

It is worth noting that prior literature reveals mixed findings regarding the influence of implementation strategy on ES adoption success. It appears that researchers tend to believe that the big-bang approach is the most risky method advised against by consultants (e.g. Markus and Tanis 2000). However, on the other hand, there are studies which illustrate a positive impact of big-bang approach. These include a study by Botta-Genoulaz and Millet (2005) who conducted research among manufacturing companies in the Rhone-Alpes region in France that have an ES "stabilised" for at least 1 year. Interestingly, they found that user satisfaction is higher in the case of the big-bang approach, as compared to the phased implementation strategy.

This study sheds light on the issue of risk connected with the adoption of the big-bang strategy. The results illustrate that this risk may be connected with lower user satisfaction. However, considering the risk of not satisfying project metrics, two kinds of gradual strategy come on the scene. Namely, the phased strategy is connected with the highest risk of exceeding planned duration time, while the parallel strategy involves the highest risk of not keeping the planned financial budget.

The examination of this study's results allows us to formulate several recommendations for practitioners dealing with ES adoptions:

- if time permits it is better to choose the phased strategy which seems to deliver the best long term effects; however, one should remember that there is a high risk of exceeding an already quite long implementation time,
- in order to minimise the project risk one should bear in mind that a project may face time-related and financial risk; in the latter case one should remember that when opting for the parallel method s/he should be prepared for problems with the budget and an excessive overload of company's finances,
- if satisfying planned project parameters is of paramount significance, it seems that the pilot or big-bang method is the best choice; however, one should be aware that it might be achieved at the expense of a lower quality of the worked out solution.

6 Conclusion

This study analyses the role of implementation strategy in ES adoption on the basis of research conducted among Polish companies. The results illustrate that the choice of implementation strategy is first and foremost influenced by project duration time and next, to a lesser extent, by company size. The main contribution of this study is that it does not limit itself to the division of implementation strategies into big-bang and gradual methods, but it performs an in-depth analysis of several types of gradual strategy. Simultaneously, it sheds new light on the influence of adopted implementation strategy on project successfulness and related risk. It suggests that risk should be considered in terms of time and finances. This study's limitations are connected with the characteristics of the sample, which contains mainly manufacturing companies introducing ES in relatively full scope. This can be a limitation in the light of current situation, when ES are adopted by companies from virtually all industries. However, on the other hand, full-scope projects have a chance of making the most of ES potential. Nevertheless, researching the specificity of various industries may be one of the avenues for future research.

References

Al-Mashari M, Al-Mudimigh A, Zairi M (2003) Enterprise resource planning: a taxonomy of critical factors. Eur J Oper Res 146:352–364
Botta-Genoulaz V, Millet P-A (2005) A classification for better use of ERP systems. Comput Ind 56:573–587
Boudreau M, Gefen D, Straub D (2001) Validation in IS research: a state-of-the-art assessment. MIS Q 25(1):1–16
Davenport TH (1998) Putting the enterprise into the enterprise systems. Harv Bus Rev 76(4):121–131
Deloitte Consulting (1998) ERP's second wave: maximizing the value of ERP-enabled processes
Duplaga EA, Astani M (2003) Implementing ERP in manufacturing. Inf Syst Manage 20(3): 68–75
El Amrani R, Rowe F, Geffroy-Maronnat B (2006) The effects of enterprise resource planning implementation strategy on cross-functionality. Inf Syst J 16:79–104
Ho C-F, Wu W-H, Tai Y-M (2004) Strategies for the adaptation of ERP systems. Ind Manage Data Syst 104(3):234–251
Holland C, Light B (1999) A critical success factors model for ERP implementation. IEEE Softw, May/June:30–36
Lui KM, Chan KCC (2005) Capability maturity model and SAP: toward a universal ERP implementation model. Int J Enterp Inf Syst 1(3):69–95
Mabert VA, Soni A, Venkataramanan MA (2000) Enterprise resource planning survey of US manufacturing firms. Prod Inventory Manage J 41(2):52–58
Mabert VA, Soni A, Venkataramanan MA (2003) Enterprise resource planning: managing the implementation process. Eur J Oper Res 146:302–314
Markus ML, Tanis C (2000) The enterprise system experience—from adoption to success. In: Zmud R (ed) Framing the domains of IT management: projecting the future through the past. Pinnaflex Educational Resources, Cincinnatti
Markus ML, Tanis C, van Fenema PC (2000) Multisite ERP implementations. Commun ACM 43(4):42–46
Newell S, Cooprider YG, David G et al (2005) Analyzing different strategies to enterprise system adoption: reengineering-led vs. quick-deployment. Int J Enterp Inf Syst 1(2):1–16

Okrent MD, Vokurka RJ (2004) Process mapping in successful ERP implementations. Ind Manage Data Syst 104(8):637–643

Olhager J, Selldin E (2003) Enterprise resource planning survey of Swedish manufacturing firms. Eur J Oper Res 146:365–373

Parr A, Shanks G (2000a) A model of ERP project implementation. J Inf Technol 1:289–303

Parr A, Shanks G (2000b) A taxonomy of ERP implementation approaches. In: Proceedings of the 33rd Hawaii international conference on system sciences, Maui, Hawaii

Rajagopal P (2002) An innovation–diffusion view of implementation of enterprise resource planning systems and development of a research model. Inf Manage 40:87–114

Robey D, Ross JW, Boudreau MC (2002) Learning to implement enterprise systems: an exploratory study of the dialectics of change. J Manage Inf Syst 19(1):17–46

Ross JW, Vitale MR (2000) The ERP revolution: surviving vs. thriving. Inf Syst Front 2(2):233–241

Sedera D, Tan FTC (2005) User Satisfaction: An Overarching Measure of Enterprise System Success. In: Proceedings of Pacific Asia Conference on Information Systems, Bangkok, Thailand, 963–976

Soja P (2007) Success factors across ERP implementation phases: learning from practice. In: Wojtkowski W, Wojtkowski WG, Zupancic J et al (eds) Advances in information systems development. New methods and practice for the networked society, vol 2. Springer, New York

Somers TM, Nelson K (2004) A taxonomy of players and activities across the ERP project life cycle. Inf Manage 41:257–278

Somers TM, Nelson K, Karimi J (2003) Confirmatory factor analysis of the end-user computing satisfaction instrument: replication within an ERP Domain. Decis Sci 34(3):595–621

Umble EJ, Haft RR, Umble MM (2003) Enterprise resource planning: implementation procedures and critical success factors. Eur J Oper Res 146:241–257

Evolution-Oriented User-Centric Data Warehouse

Darja Solodovnikova and Laila Niedrite

Abstract Data warehouses tend to evolve, because of changes in data sources and business requirements of users. All these kinds of changes must be properly handled, therefore, data warehouse development is never-ending process. In this paper we propose the evolution-oriented user-centric data warehouse design, which on the one hand allows to manage data warehouse evolution automatically or semi-automatically, and on the other hand it provides users with the understandable, easy and transparent data analysis possibilities. The proposed approach supports versions of data warehouse schemata and data semantics.

1 Introduction

Data warehouses are databases designed for querying and analysing data. The main goal of a data warehouse is to provide the most accurate and historically correct information to users in the most convenient and easy understandable way to support the analysis of business processes and decision making. On the one hand, this means that the data warehouse must reflect all the changes that occur in the analyzed process, and on the other hand users should receive answers to their questions as fast and easy as possible.

Data warehouses integrate information from various data sources that can change in the course of time. Besides, business requirements often evolve at the client level. To reflect this evolution, it is possible to adapt the existing data warehouse schemata and data extraction, transformation and loading (ETL) processes, but this solution can cause a loss of history. This is why it is preferable to keep track of the evolution. This can be realized by data warehouse schema versions. According to [1], 'schema version is a schema that reflects the business requirements during a

D. Solodovnikova · L. Niedrite (✉)
University of Latvia, Raina bulv. 19, 1586 Riga, Latvia
e-mail: darja.solodovnikova@lu.lv

L. Niedrite
e-mail: Laila.Niedrite@lu.lv

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_58,

given time interval, called its validity, that starts upon schema creation and extends until the next version is created'.

In this paper we propose the approach to supporting data warehouse evolution, creation and management of data warehouse schema versions and versions of data semantics, analysis of data warehouse data by users in easily designed and modified reports that take into account existence of data warehouse versions.

The rest of this paper is organized as follows. In Sect. 2 the related work is presented. In Sect. 3 the proposed data warehouse framework is outlined. The Sect. 4 describes the working example used throughout the rest of the paper for demonstration purposes. The main contribution of this paper is presented in Sects. 5 and 6, where the metadata model for a multiversion data warehouse is described and different aspects of creation and execution of reports are discussed.We conclude with directions for future work in Sect. 7.

2 Related Work

In the literature there are various solutions for the data warehouse evolution problems. In [2] dimension update operators are formally specified, and their effect is studied over materialized views over dimension levels. In [3] the primitive evolution operations that occur over the data warehouse schema are defined. In [4] the generalized data warehouse model formally is defined, which supports extended hierarchies, and the transformation rules of evolution operators are specified.

The above mentioned papers do not address the problems of the data warehouse adaptation after changes in data sources. Several approaches have been proposed for solving these problems [5, 6]. These approaches are based on mappings or transformations that specify how one schema is obtained from the other schema. This specification is used to adapt one schema after changes in the other schema.

In several papers [7, 8] a data warehouse is defined as a set of materialized views over data sources. These papers study the problems of how to rewrite a view definition and adapt view extent after changes in source data and schemata.

Several authors [1, 9, 10] propose the data warehouse schema versioning approach to solve the problems of schema evolution. The main idea in [1] is to store augmented schemata together with schema versions to support cross-version querying. Though the metadata of schema versions is mentioned, the details are not explored. In [10] a method to support data and structure versions of dimensions is proposed. The method allows tracking history and comparing data using temporal modes of presentation that is data mapping into the particular structure version. In [9] metadata management solutions in a multiversion data warehouse are proposed. Issues related to queries over a multiversion data warehouse are considered in [11], but the translation of queries to SQL is not discussed.

The above mentioned papers consider only one kind of evolution problems, for example, changes in a data warehouse schema raised by evolving business requirements, changes in data sources or data warehouse. In our approach we propose the solution that is able to handle all these kinds of evolution problems. Besides there was too little research on implementation methods of data warehouse evolution and creation and execution of reports on multiple data warehouse schema versions.

3 Data Warehouse Evolution Framework

To handle data warehouse evolution problems, we propose the data warehouse framework. The detailed description of the framework is given in [12]. The framework is composed of the development environment, where the metadata repository is located and ETL processes and change processing is conducted, and the user environment, where reports on one or several data warehouse versions are defined and executed by users.

All operation of the data warehouse framework is based on the metadata, which are used to describe the data warehouse schema versions, their storage in the relational database and semantics of data stored in the data warehouse, and to accumulate information about reports defined by users on schema versions.

The data warehouse evolution framework supports physical, logical and semantic changes that create a new data warehouse schema version. Physical changes operate with database objects (tables, columns), but logical and semantic changes modify mainly schema metadata. The examples of physical changes are renaming, creation or deletion of an attribute, measure, dimension or cube, etc. The examples of logical changes are connection or disconnection of a dimension from a cube, creation or deletion of a hierarchy or level, etc. Semantic changes refer to adaptation of meanings of the same data objects. As a result of a logical change, logical metadata are adapted and there may be no changes at the physical level, except for new keys or key columns. As a result of a physical change, both logical and physical metadata are modified. As a result of semantic change usually the semantic and logical metadata are adjusted.

In this paper we will concentrate on the part of the framework related to the metadata models of data warehouse versions, execution of reports and analysis of data warehouse data.

4 Working Example

Let us consider a working example of a data warehouse, which was designed at our university to store data about course usage in the e-learning environment. The evolution of the data warehouse is depicted in Fig. 1. The first version V_1 of the data warehouse was created on 1 Sept 2004. It contained a cube Activity, where the number of hits, total usage time in hours, and number of active users was accumulated. It was possible to analyze these measures by the course, tool, user role in a course and time.

Using this data warehouse, it became obvious that its' granularity was not satisfactory and on 1 Feb 2005 the version V_2 was created, where the activity was collected

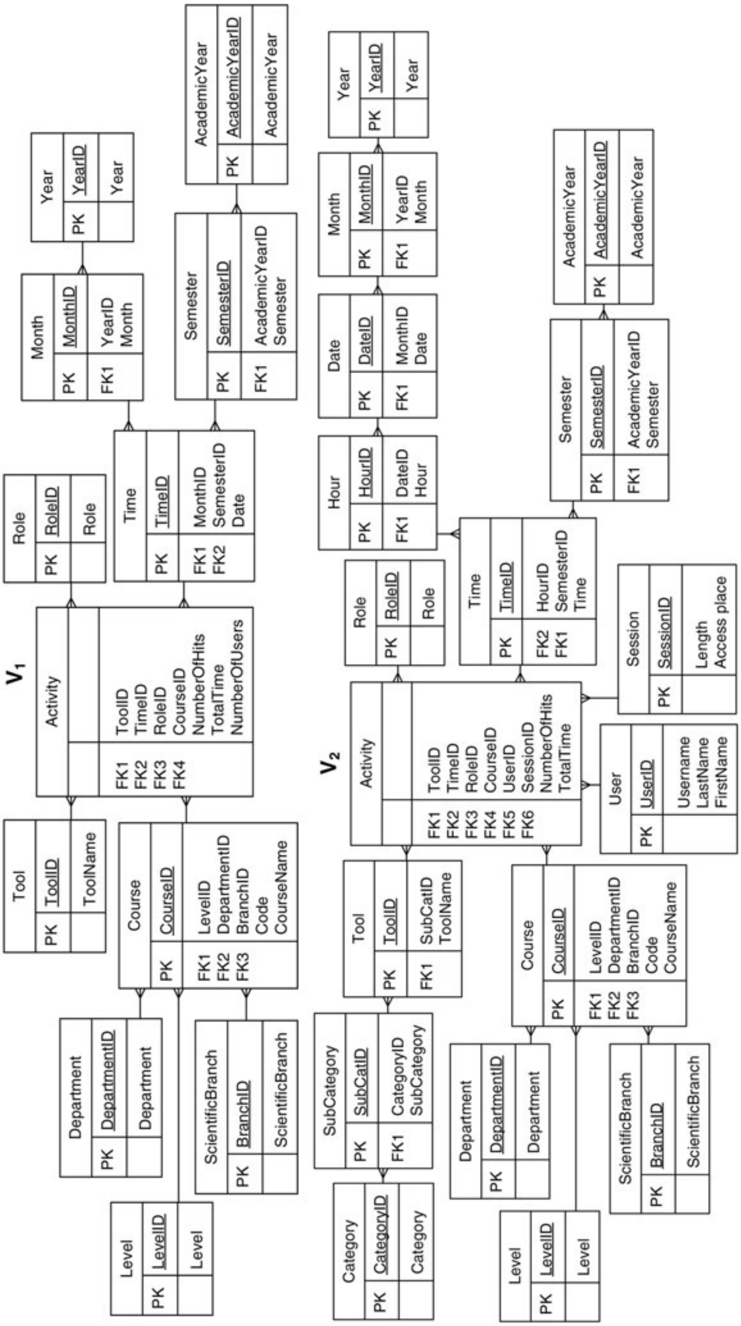


Fig. 1 Data warehouse schema versions

separately for each user and session. The data warehouse was redesigned and two data warehouse versions were created.

The following changes were conducted to implement the new schema version of the data warehouse:

- Physical changes:
  - Addition of new dimensions User and Session;
  - Addition of new dimension attributes Time and Hour of the Time dimension, SubCategory and Category of the Tool dimension;
  - Deletion of measure NumberOfUsers from the Activity cube.
- Logical changes:
  - Connection of the User and Session dimensions to the Activity cube;
  - Addition of new hierarchy ToolHierarchy to the Tool dimension;
  - Addition of new levels ToolName, SubCategory and Category to the ToolHierarchy; levels Time and Hour to the TimeHierarchy;
  - Connection of attributes ToolName, SubCategory, Category, Time and Hour to the appropriate hierarchy levels.
- Semantic change—Change of meaning of the measure TotalTime from 'Usage time in hours' to 'Usage time in seconds'.

5 Metadata Repository

The metadata repository of the data warehouse evolution framework describes a data warehouse at three levels, namely logical, physical and semantic levels. Besides the repository also contains reporting metadata, which describes the structure of reports generated by users.

Common Warehouse Metamodel (CWM) [13] was used as a basis of the proposed metamodel of multidimensional data warehouse. CWM consists of packages, which describe different aspects of a data warehouse. In the next sections each type of metadata is outlined.

5.1 Logical Metadata

Metadata at the logical level describe the multidimensional data warehouse schema. The logical level metadata are based on the OLAP package of CWM and contains the main objects from this package, such as dimensions and cubes connected by cube-dimension associations, measures, attributes, hierarchies, etc.

To reflect multiple versions of a data warehouse schema, two objects were introduced: SchemaVersion and VersionTransformation (Fig. 2), which are not included in CWM. An object SchemaVersion corresponds to a data warehouse schema version, which is created as a result of some change in a data warehouse schema. Each schema version has a validity period defined by attributes ValidFrom and ValidTill. Each version, except for the first one, has a link to a previous version.

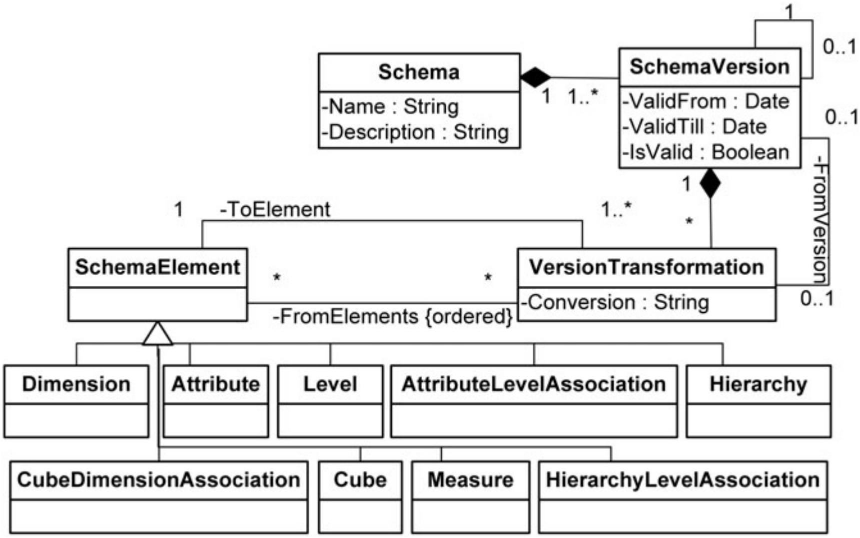


Fig. 2 Schema versions in logical metadata

Elements of a version are connected to the SchemaVersion by the VersionTransformation. The association ToElement connects an element of the current version. A Schema Element can be any element of the logical metamodel, for example, Measure, Cube, etc. If an element remains unchanged it is connected to several versions through the VersionTransformation. The attribute Conversion stores a function that obtains a changed element from elements of other version. Elements of other version, which are used to calculate the changed element of a new version, are connected to VersionTransformation by the association FromElements and the corresponding version is connected by the association FromVersion.

For the example data warehouse (Fig. 1) in the logical metadata, two schema versions V_1 and V_2 are constructed. Both schema versions include dimensions, cubes, attributes, measures, hierarchies and corresponding associations, according to the schemata in Fig. 1. The elements, which are common to both versions, are connected by version transformations with empty conversion functions to versions V_1 and V_2. The measure NumberOfUsers is connected only to version V_1. Since the granularity of other measures is different in both versions, in the logical metadata the two different measure objects are constructed for each of the measures NumberOfHits and TotalTime and connected to each of the versions. The dimensions User and Session and the corresponding cube-dimension associations between these dimensions and cube Activity are connected only to version V_2. Also the new attributes of the dimensions Time and Tool and corresponding hierarchies made of these attributes exist only in the new version V_2.

The version transformations given in Table 1 are generated for measures. The transformations are created from version V_2 to version V_1, because the transformations of measures in other direction are not possible. Also transformations are possible for new Tool dimension attributes Subcategory and Category.

Table 1 Version transformations

FromVersion	ToVersion	ToElement	Conversion
V_2	V_1	NumberOfUsers	COUNT(DISTINCT Activity.UserID)
V_2	V_1	NumberOfHitsV1	SUM(Activity.NumberOfHitsV2)
V_2	V_1	TotalTimeV1	SUM(Activity.TotalTimeV2/3600)

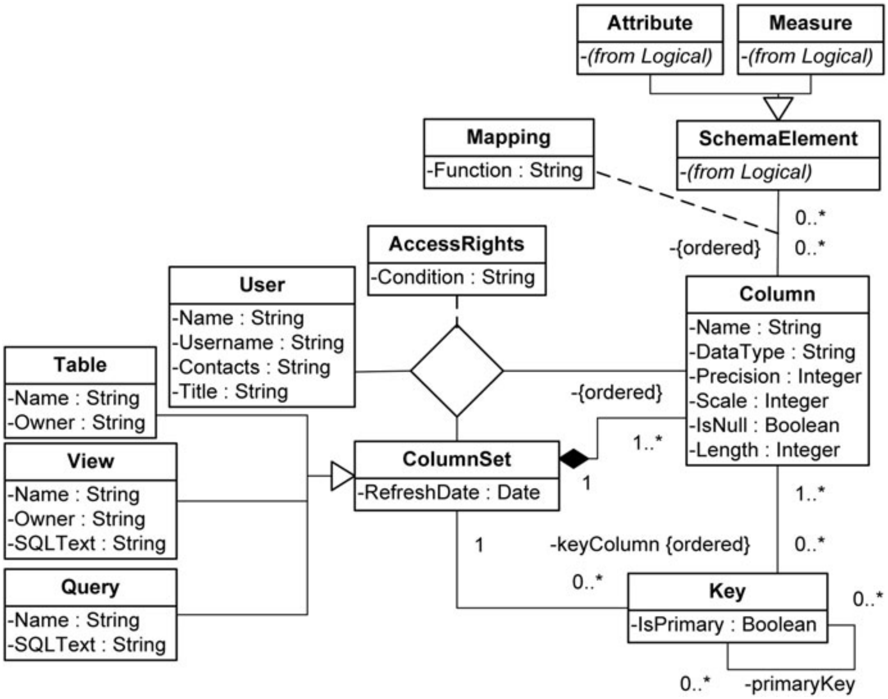


Fig. 3 Physical metadata

5.2 *Physical Metadata*

Metadata at the physical level describe relational database schema of a data warehouse and mapping of a multidimensional schema to relational database objects from the logical level. The model of physical level metadata is shown in Fig. 3. Physical metadata do not include versioning information because in the database there is only one schema version and versioning is implemented at the logical level. The physical level metadata are based on the Relational package of CWM. The objects of physical and logical levels are connected by objects defined in the Transformation package of CWM. This means that attributes of dimensions and measures of cubes are defined by Mappings, which specify formulas that obtain attributes and measures from one or several columns of physical tables and views.

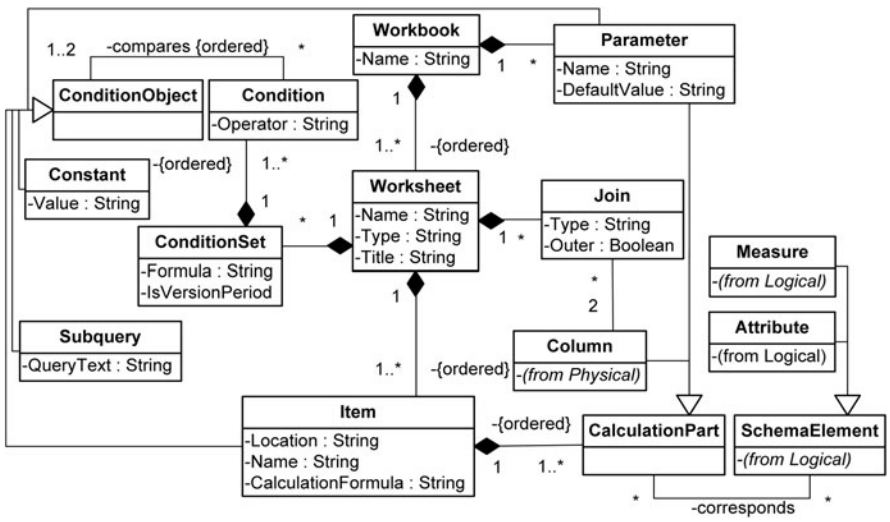


Fig. 4 Reporting metadata

For the example data warehouse physically in the database all schema versions are stored in one physical schema. The schema of the first version of the data warehouse is created according to the model V_1 in Fig. 1. When the version V_2 is created, the new tables User and Session are created. In these tables new fictive records with data "All together" are created with an identifier U and S respectively. The columns User_ID and Session_ID are added to the table Activity and filled with U and S for all existing data. The table Time and Tool are supplemented with columns Time, Hour and Subcategory, Category respectively. The new columns Subcategory and Category are updated with data for the existing records in tables to form corresponding hierarchies, because these columns correspond to new hierarchy levels of smaller detail. But the new columns of the Time dimension correspond to new hierarchy levels of grater detail, therefore, they are updated with fictive data, for example, time '00:00'. The column NumberOfUsers is not removed, but is no longer updated by ETL processes. For the new versions of other two measures additional columns are not created, i.e. both versions of each measure are stored in one table column.

5.3 *Reporting Metadata*

Reporting metadata describe structure of reports. CWM contains a package Information Visualization that describes how data warehouse elements are rendered. These metadata are very general and not sufficient, therefore, a new metamodel (Fig. 4) was developed. Basically reports are worksheets that contain data items defined by calculations, which specify computation formulas from parameters and table columns, which usually correspond to schema elements. Reports also consist of user-defined conditions and joins between tables.

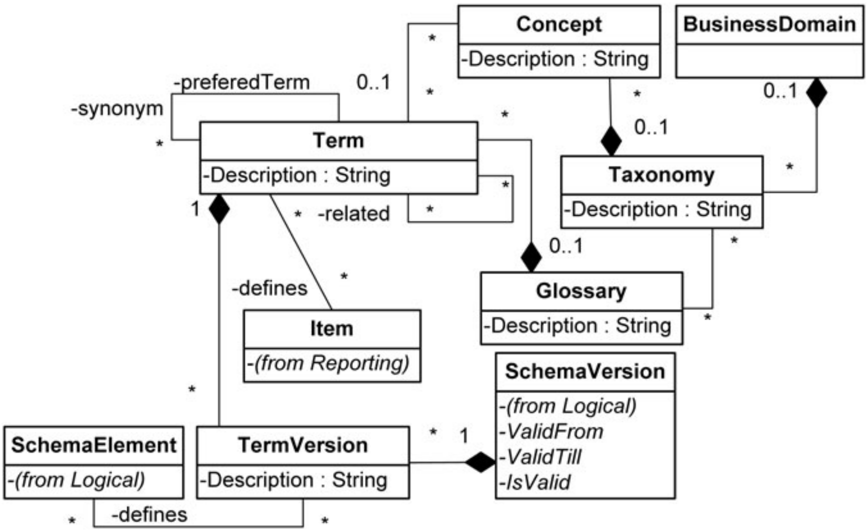


Fig. 5 Semantic metadata

5.4 Semantic Metadata

Data warehouse users must understand the semantics of data that appear in reports from business perspective. They also must be able to analyse these data using all necessary features, including OLAP operations drill-down and roll-up, using hierarchies. Besides, it is desirable that users can modify or construct reports themselves from elements, which are familiar to them, so that reports creation becomes transparent. For these purposes, it is necessary to describe each element of the data warehouse model in business language. This description could also be used by users to express their requirements for information and changes in requirements making the understanding between users and developers of data warehouse clearer. The description of data warehouse elements in business language is stored in semantic metadata.

In CWM there is the package Business Nomenclature, which can be used to represent business metadata. This package was used as a basis for semantic metadata depicted in Fig. 5. The main classes that are used for description of data warehouse elements are Terms and Concepts, which are united in Glossaries and Taxonomies respectively. A concept is the semantic meaning or notion of some data warehouse element or data stored in some element, but a term is particular word or phrase used by users to refer to a concept. Terms define items used in a report. There may be preferred terms and synonym terms to identify a concept. Also terms may be related to each other.

The CWM metamodel was supplemented with the class TermVersion, which is connected to the particular schema version. The new term version reflects the meaning of some data warehouse schema element, which was valid during the particular period of time defined by attributes ValidFrom and ValidTill of the corresponding schema version.

Different versions of terms may be created in case of changes of the data warehouse business requirements. For example, in the working example data warehouse the measure TotalTime is defined by two term versions ‘Total time of all users in hours’ and ‘Total time of each user in seconds’.

6 Reports on Multiversion Data Warehouse

In the evolution data warehouse framework reports are constructed by users, who select desirable terms and term versions from the semantic metadata. The structure of concepts and terms is displayed as a graph, which shows taxonomies of concepts, their associated terms, relations between terms and term versions. The report items, which correspond to selected terms and term versions, are automatically created in the reporting metadata. Users also can define new report items and their corresponding terms, which are saved in the semantic metadata and become available to other users. Users must also specify the location of selected terms in the report. There is also a possibility to define report parameters and conditions. All necessary reporting metadata are created automatically, according to requirements of users.

Example Report. To demonstrate the execution of reports in the following sections, according to the approach presented in the paper, let us use the most popular report executed in the example data warehouse. This report was first developed in the first version of the data warehouse schema. The report is widely used by lecturers of particular courses, who are interested in the detailed analysis of the usage of taught courses. It displays the number of students (NumberOfUsers), who used the specific tool (ToolName) in their course during some period of time and the duration of this tool usage (TotalTimeV1).

6.1 Building Queries on Multiple Versions

6.1.1 Version Selection

When a user runs a report, at first the list of data warehouse versions, which contain report items, is obtained.For each attribute or measure used in the report, all versions that contain it and that were valid in the time period of the report are collected from the logical metadata. If only one version remains then report data are presented according to this version. If more then one version remains, further analysis is necessary.

To determine options of report data presentation, a special relationship matrix is being constructed. The columns of the matrix correspond to versions, but rows correspond to schema elements, selected for the report. If a schema element exists in a version (it is obtained by a version transformation without conversion), then ‘1’ is recorded in the corresponding matrix cell. If a schema element does not exist in a version, but it is obtainable with the transformation version by conversion function from other elements, then the corresponding matrix cell is filled with ‘2’. If a schema

a

Schema Element	Version V$_1$	Version V$_2$
CourseName (Course)	1	1
TotalTimeV1 (Activity)	1	2
ToolName (Tool)	1	1
Date (Time)	1	1
NumberOfUsers (Activity)	1	2

b

Schema Element	Version V$_1$	Version V$_2$
CourseName (Course)	1	1
TotalTimeV1 (Activity)	1	2
ToolName (Tool)	1	1
Date (Time)	1	1
NumberOfUsers (Activity)	1	2
SubCategory (Tool)	2	1

Fig. 6 Relationships matrix

element does not exist in a version and can not be obtained with any conversion function (no version transformation), then in the corresponding matrix cell '0' is recorded. Depending on the values of cells of the relationship matrix, the following options of report presentation are available:

1. The report can be presented *in accordance with one particular version*, if all matrix cells of the column, which corresponds to that version, are filled with '1' and all other cells are filled either with '1' or '2'.
2. If neither of versions does contain all elements, then *elements from different versions can be presented in one report*, if all matrix cells contain '1' or '2'.
3. If any matrix cell contains '0', then the report can only be displayed *separately for each version*.

If the example report is executed for the time period that spans both schema versions, then the corresponding matrix is depicted in Fig. 6a, where it can be seen that the report can be presented according to the first version. If we change a little the definition of the report to display data by tool Subcategory, the report then can be presented only with elements from various versions (Fig. 6b).

6.1.2 Query Generation

When a user chooses any option of report presentation, an SQL query is built based on a report definition in reporting metadata, and its result is displayed to a user. An SQL query is constructed, according to special algorithm consisting of the following steps:

1. Analysis of chosen items and determination of used column sets;
2. Analysis of joins;
3. Generation of list of conditions;
4. Grouping and construction of conditions with aggregates functions;
5. Adding restrictions of user rights;
6. Simplification and optimization of the query;
7. Supplementation of a query with version transformations.

The details of the algorithm were published in the paper [14]. According to the query construction algorithm, the query displayed in Fig. 7 is constructed for the example report.

```
/*Q1*/SELECT TOOL_NAME, DATE1, SUM(NUMBER_OF_USERS), SUM(TOTAL_TIME)
FROM COURSE, TOOL, TIME, ACTIVITY, ROLE
WHERE ROLE.ID=ROLE_ID AND COURSE.ID=COURSE_ID AND TOOL.ID=TOOL_ID AND TIME.ID=TIME_ID
AND DATE1 BETWEEN TO_DATE(:"from",'dd.mm.yyyy') AND TO_DATE(:"until",'dd.mm.yyyy')
AND DATE1 BETWEEN TO_DATE('01.09.2004','dd.mm.yyyy') AND
TO_DATE('31.01.2005','dd.mm.yyyy')
AND COURSE_NAME=:"course" AND ROLE.ROLE='Student' GROUP BY TOOL_NAME, DATE1
UNION
/*Q2*/SELECT TOOL_NAME, DATE1, COUNT(DISTINCT USER_ID), SUM(TOTAL_TIME)
FROM COURSE, TOOL, TIME, ACTIVITY, ROLE
WHERE ROLE.ID=ROLE_ID AND COURSE.ID=COURSE_ID AND TOOL.ID=TOOL_ID AND TIME.ID=TIME_ID
AND DATE1 BETWEEN TO_DATE(:"from",'dd.mm.yyyy') AND TO_DATE(:"until",'dd.mm.yyyy')
AND DATE1 BETWEEN TO_DATE('01.02.2005','dd.mm.yyyy') AND sysdate
AND COURSE_NAME=:"course" AND ROLE.ROLE='Student' GROUP BY TOOL_NAME, DATE1;
```

Fig. 7 Query on multiple schema versions

6.2 Using Hierarchy Versions

Data warehouse reports should support high interactivity with a user including OLAP operations: roll-up, drill-down, etc. This is why the reporting tool allows to transform data in the report according to user needs. One of the possibilities to analyze data from different points of view is to use hierarchies defined in the logical metadata. In the metadata, different versions of hierarchies, levels and associations between attributes and hierarchy levels can be created. When a user runs a report, the query construction algorithm also identifies all hierarchies and their structure that exist in the particular data warehouse version, which is selected by a user to present report data. If a user chooses to present elements from different versions in one report, then only hierarchies, which exist in all versions, are available as well as hierarchies that can be transformed by version transformations.

For the example report executed in accordance with the first version, the following hierarchies are available: all Course dimension hierarchies, Time hierarchy, which consists of only three levels Date, Month and Year. But if the report includes tool Subcategory, then additional Tool hierarchy can be used to analyse data, because it exists in the second version and the version transformation can be constructed to map lower hierarchy level ToolName to higher hierarchy levels.

6.3 Using Term Versions

In reports term versions are used to separate different meanings of the same schema element. If any schema element has multiple term versions, when a user runs a report, which includes this schema element, he is informed that for the same schema element two term versions exist. The user has to choose the preferred term version and then the appropriate schema element version is included in the report.

In the example report, the measure TotalTime was used. The semantics of this measure is different in two versions of the data warehouse. In the first version TotalTime was accumulated in hours and summarized for all users of the e-learning environment, but in the second version this measure was accumulated in seconds for each user and session. So to execute the example report the user must select one of two term versions ‘Total time of all users in hours’ or ‘Total time of each user in seconds’ and the appropriate data are presented in the report.

7 Conclusions and Future Work

In this paper we have presented an approach to data warehouse design. This approach is evolution-oriented because it supports adaptation of data sources and business requirements and allows to propagate different changes in data warehouse creating versions of schemata and data semantics. This approach is user-centric, because users themselves can design reports on multiple data warehouse versions using terms, which are familiar to them. Besides changes in users’ requirements are taken into account.

Several directions for future research related to the presented issue are personalization of reports built on multiple data warehouse versions and support of automatic adaptation of data warehouse according to user needs expressed using terms and term versions from semantic metadata without or with minimal participation of a data warehouse administrator.

Acknowledgements This work has been supported by ESF projects No. 2009/0216/1DP/1.1.1.2.0/09/APIA/VIAA/044 and 2009/0138/1DP/1.1.2.1.2/09/IPIA/VIAA/004.

References

1. Golfarelli M, Lechtenbörger J, Rizzi S, Vossen G (2006) Schema versioning in data warehouses: enabling cross-version querying via schema augmentation. Data Knowl Eng 59(2):435–459
2. Hurtado CA, Mendelzon AO, Vaisman AA (1999) Maintaining data cubes under dimension updates. In: 15th international conference on data engineering. IEEE Computer Society, Sydney, pp 346–357
3. Blaschka M (2000) FIESTA: a framework for schema evolution in multidimensional databases. PhD thesis, Technische Universitat Munchen, Germany
4. Banerjee S, Davis KC (2009) Modeling data warehouse schema evolution over extended hierarchy semantics. J Data Semant XIII, LNCS 5530. Springer, Heidelberg, pp 72–96
5. Marotta A (2000) Data warehouse design and maintenance through schema transformations. Master thesis, Universidad de la República Uruguay
6. Velegrakis Y, Miller RJ, Popa L (2003) Mapping adaptation under evolving schemas. In: 29th international conference on VLDB. Morgan Kaufmann, Berlin, pp 584–595
7. Bellahsene Z (2002) Schema evolution in data warehouses. Knowl Inf Syst 4:283–304
8. Rundensteiner EA, Koeller A, Zhang X (2000) Maintaining data warehouses over changing information sources. Commun ACM 43(6):57–62
9. Wrembel R, Bebel B (2005) Metadata management in a multiversion data warehouse. In: OTM conferences (2), LNCS 3761. Springer, Heidelberg, pp 1347–1364
10. Body M, Miquel M, Bedard Y, Tchounikine A (2002) A multidimensional and multiversion structure for OLAP applications. In: ACM 5th international workshop on data warehousing and OLAP. ACM, McLean, pp 1–6

11. Morzy T, Wrembel R (2004) On querying versions of multiversion data warehouse. In: ACM 7th international workshop on data warehousing and OLAP. ACM, Washington, pp 92–101
12. Solodovnikova D (2007) Data warehouse evolution framework. In: Spring young researcher's colloquium on database and information systems, Moscow, Russia
13. Object Management Group (2003) Common warehouse metamodel specification, v1.1 http://www.omg.org/cgi-bin/doc?formal/03-03-02
14. Solodovnikova D (2008) Building queries on multiple versions of data warehouse. In: 8th International Baltic conference on DB&IS, Tallinn, Estonia, pp 75–86

Change Management for Fractal Enterprises

Renate Strazdina and Marite Kirikova

Abstract Change management is an important process enabling the definition of a successful enterprise strategy and operations—especially in a turbulent environment. A number of methodologies are available for change management; however none of those is designed specifically for fractal systems. Taking into consideration self-similarity; self-organization, goal-orientation, and dynamics and vitality of fractal systems, change management in such systems obviously requires organizational procedures that take account of the distinct properties of fractal systems. The purpose of this research is to define a change management methodology that would allow administering change management in fractal systems in general and apply it to a specific change situation when the primary object of the change is an information system of a fractal enterprise.

1 Introduction

Change management is an important issue—especially in a turbulent environment. It can involve changes both to the organizational structure and processes as well as information system changes that are the next step after the former two categories of changes [1]. The way changes are introduced into enterprises should improve organizational functionality while preserving the essential qualities of the enterprise. The quality we focus on in this paper is the fractality of an enterprise. Fractal systems (enterprises) are considered a viable option of organizational structure and operation in unpredictable and rapidly changing environments [2–5]. A number of approaches are available for change management [1, 6–8], however none of those is designed specifically for fractal systems. The purpose of this research is to define a change management process that would allow managing change in fractal systems in general and, in particular, in cases of changes in information systems of fractal enterprises.

R. Strazdina · M. Kirikova (✉)
Riga Technical University, Riga, Latvia
e-mail: renate.strazdina@cs.rtu.lv

M. Kirikova
e-mail: marite.kirikova@cs.rtu.lv

J. Pokorny et al. (eds.), *Information Systems Development,*
DOI 10.1007/978-1-4419-9790-6_59,

To achieve the above-mentioned purpose, the following tasks were set:

- Identify those qualities of fractal systems that should be preserved throughout the change management process;
- Consider the existing change management methodologies in order to select the core elements of the change management process to be applied to fractal systems—principles, procedures and examples of application thereof; and
- Adapt the core elements of change management to the specific requirements of fractal systems.

The paper is organized as follows: Section 2 describes fractal systems; Sect. 3 describes the change management process and presents an analysis of the existing change management approaches; Sect. 4 defines the core elements of the change management process for fractal systems; Sect. 5 shows the results of applying the previously defined process to changes in information systems; Sect. 6 describes the main conclusions and future research questions.

2 Fractal Systems

A fractal system consists of self-similar and relatively autonomous units/components or fractals that are the elements of the system. A fractal system is an interactive system that can adapt to a changing environment as well as to new goals [9]. The following core qualities of fractal systems enable them to change according to internal and external requirements [8]:

- *Self-similarity*: Fractals share inputs and outputs but each has its own particular internal structure;
- *Self-organization:* The ability to adjust the behavior of a fractal and its mechanisms for interaction with other fractals to achieve the overall goals of the system. This enables fractals to restructure, renew and split;
- *Goal-orientation:* A fractal's goals are set by coordinating processes with other fractals—and the goal can be modified if necessary. The goals of the system are achieved through a number of iterations, by modifying an individual fractal's goals and taking account of its feedback loop(s); and
- *Dynamics and vitality:* Coordination and cooperation between self-organizing fractals is characterized by individual dynamics and the ability to adapt to a dynamically changing environment.

A fractal, if a functional unit of the enterprise, has unique goals that achieve specific results and works autonomously in a self-organizing mode—as well as interacts with other fractals [9, 10]. To preserve the fractality of a system it is necessary to ensure that after the changes all the above-mentioned qualities are still present in the enterprise.

The internal structure of a fractal can be described from the systems theory point of view which states that every system consists of subsystems that are in mutual

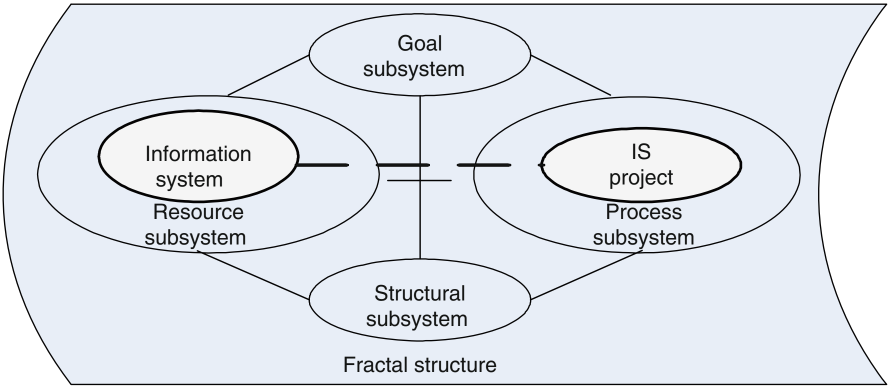


Fig. 1 The structure of a fractal

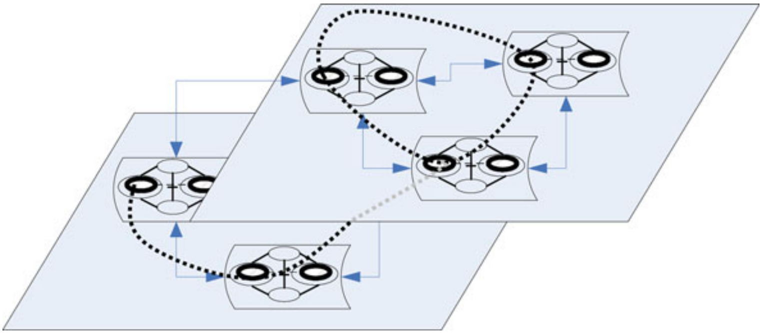


Fig. 2 The structure of the fractal system and coordination between fractals (lines with *arrowheads*) at different functional levels of the system and relationship between information systems of different fractals (*dotted lines*)

interaction (goals subsystem, structure subsystem, resource subsystem, and process subsystem), where changes in one subsystem may lead to changes in other subsystems [11]. In case of a business organization an important subsystem of the enterprise fractal's resource subsystem is its information system (IS) (See Fig. 1) and the subsystem of the process subsystem is the IS project [12]. Changes in other subsystems can cause changes in IS and the IS project and vice versa.

Thus it is possible to conclude that changes in the enterprise fractal can cause changes in the information system deployed therein. So by planning a change management process in fractal systems, information system changes in every fractal should be taken into consideration. As coordination and cooperation between self-organizing fractals is one of the core qualities of the fractal system (See Fig. 2), when planning change processes it should be taken into consideration that changes in the information system of one fractal can cause changes in the information systems of other fractals.

3 Related Work on Change Management

Change management is a structured approach to moving from the current state to a future state. Change management is interpreted differently—from continuous change that involves, say, minor yet constant changes in existing processes to significant, pervasive changes that might involve, for example, amendments to the organization's strategy. Furthermore, the change management process is different for organizations, individuals and technological systems. This research only looks at change management at the organizational level.

The following are the common basic steps of change manage process according to [3]:

- Identification of the necessary changes
- Definition of the necessary outcome
- Planning of activities for achieving the change
- Performing the activities
- Maintaining the changes made

These steps will be taken as the basic components of the change management process proposed in the next section. Change management may be performed in different styles. In accordance with Boddy and Buchanan [4] there are three management styles:

- Imposition
- Education and communication
- Participation

Imposition is possible when a manipulative approach to change management is practiced while the other two change management styles require particular support structures for change management [1].

By taking into consideration the core qualities of the fractal systems, the most appropriate styles are *education and communication* as well as *participation* since imposition would require an organizational structure that allows issuing imperative orders.

The following parameters are also important for the change management process:

- Scope of change management—minor changes/changes in specific areas of the organization (e.g., information systems changes) *or* changes that affect the entire organization;
- Nature of change management—reactive *or* proactive; and
- Cause of change management—changes in internal *or* external environment.

Depending on the change management parameters' values, different methods can be applied in the change management process. This observation refers both to regular and fractal systems. Change management performed in fractal systems has to be performed taking into consideration also other common issues relevant in change management in general:

- Formulating a change management strategy (maturity assessment);
- Engaging top management representatives—change managers (sponsors);
- Developing the skills and knowledge to support the changes (education and training);
- Assisting the organization's employees in the change process (support from change managers); and
- Ensuring the viability of changes (system of metrics).

While it is important to consider the above-mentioned issues in change management for fractal systems, the main emphasis should still be placed on preserving the fractal qualities of the enterprise. In the next section we propose a change management process where for each of its steps the fractal qualities of the enterprise are considered in order to sustain the viability of the system throughout the change process.

4 Change Management Process for Fractal Systems

Since a fractal system is a system that can adapt to a changing external environment, a change management process is "built in" in it—otherwise a fractal system does not have an adaptive mechanism available. As mentioned in the previous sections, the following core qualities of a fractal system ensure its capacity to adapt: self-similarity, self-organization, goal-orientation, and dynamics and vitality.

Accordingly, the change management process must take these qualities into account. As such, the basic steps of the change management process need to include the following amendments (see Table 1):

Table 1 shows that one of significant aspects of change management process in fractal systems is not only the support of enterprise top managers in the change management process—but also the support of each fractal's management.

5 An Example of Change Management Methodology Applied to a Fractal Enterprise

This research uses an example of an introduction of a new IS in an international company that can be viewed as a fractal enterprise (Fig. 3).

The company operates globally, following the same principles—e.g., company brand, risk and quality management systems, information systems, etc. Each division based in its respective country of operation is a separate legal entity—however, all such entities follow the same organizational structure across all the locations. That allows sharing support functions that serve an entire region while being based in a single territory. A Senior Partner is in charge of each legal entity; each such entity consists of at least four business units that are led, respectively, by a partner (one of such partners can be the Senior Partner for the respective territory). Legal entities across multiple countries are grouped into a region managed by a Regional Partner,

Table 1 Change management steps for fractal systems

Basic steps of change management	Change management steps for fractal systems
Identification of the necessary changes	
Identification of the necessary changes in strategy, processes, structure, knowledge, information systems in the entire organization or in specific parts thereof	*Self-similarity:* It is necessary to identify whether the changes are needed in the entire organization or in the particular fractal *Self-organization:* Changes can be initiated: • In a particular fractal, without changes in the entire system • In order to achieve better co-operation among the fractals • It is necessary to take into account the existing adaptation mechanisms when changes in the entire system are identified *Goal-orientation:* When identifying changes it should be taken into account that every fractal is goal-oriented and it is possible that changes impact either the goals of the fractal or the goals of the entire system *Dynamics and vitality:* When identifying changes it should be taken into account that fractal systems have a core quality—Coordination and cooperation between self-organizing fractals. Thus it is possible to conclude that changes that are identified in one fractal can theoretically cause changes in other fractals. So it is necessary to identify all the existing relationships among the fractals
Definition of the necessary outcome	
Definition of the result to be reached by carrying out the change management process	*Self-similarity:* The defined result should be robust in order to be applicable to different structures of fractals *Self-organization:* No specific activities identified for fractal systems *Goal-orientation:* The result should be goal-oriented and if the change process causes changes in the goals of a fractal it should be taken into consideration *Dynamics and vitality:* When defining the result, co-operation between fractals should be taken into consideration
Planning of activities for achieving the change	
Activities shall include at least the following sub-activities: • Development of change management strategy • Engaging top-level management • Development of knowledge and skills to support the change management process • Support of employees during the change management process • Ensuring change sustainability • Any specific activities for particular change management case	*Self-similarity:* Planned activities can be modified according to the needs of the particular fractal. They can be supplemented with additional activities. Therefore it is necessary to define the goals to be achieved precisely, in order to "integrate" them with the goals of the other fractals *Self-organization:* Basic activities "are to be managed from the top", for example by providing the methodology at the same time permitting the changes needed for every fractal *Goal-orientation:* No specific activities identified for fractal systems *Dynamics and vitality:* Activities should be planned so that all the relationships existing among fractals are taken into consideration

Table 1 (continued)

Basic steps of change management	Change management steps for fractal systems
Performing the activities	
Performing the activities according to defined plan	*Self-similarity:* Defined activities should be robust, because they are to be applied in every fractal according to the structure of the fractal. So it is necessary to define the results precisely in order for all fractals to achieve the same results *Self-organization:* By defining precisely the activities to be performed and results to be achieved fractals will self-organize themselves in order to achieve the defined results *Goal-orientation:* If all activities to be performed are defined, fractal as a self-organizing system can modify the existing goals and processes in order to attain the newly set goals. The open question is regarding the fractal's desire/need to make changes in its goals. In order to do this there is a need for involvement and support from the fractal's management and fractal's employees *Dynamics and vitality:* Co-ordination of activities performed should be carried out in conjunction with other fractals
Maintaining the changes made	
A procedure for monitoring the implemented changes is needed. The procedure also describes activities to be performed in case sustainability is somehow jeopardized	Fractal system with its core qualities is tailored for ongoing change management, so no specific activities identified for fractal systems

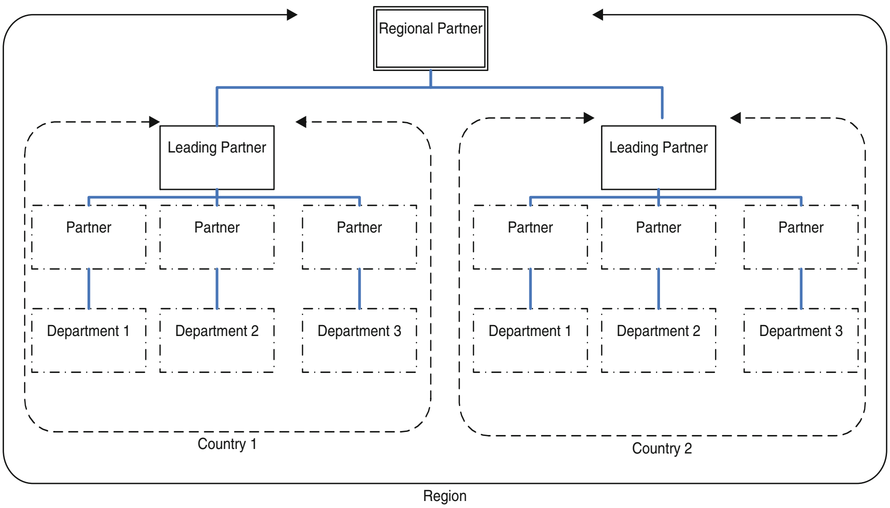


Fig. 3 An example of the fractal enterprise

while the regions make up the entire company managed by Global Partners. In addition, there are also units grouped by business line—both regionally and globally.

This company can be viewed as a fractal enterprise since it possesses all of the qualities of fractal systems:

- *Self-similarity:* All legal entities have identical requirements in terms of goals that need to be achieved and the procedures to be used—yet at the same time each country or department can modify these procedures to reflect the local market conditions or legislation requirements, as long as the minimum conditions mandated by the company procedures are met.
- *Self-organization:* The company adapts to the necessary changes in the external environment and, if necessary, cooperates with the legal entities in other territories—or departments within a single entity cooperate if this is necessary to meet the needs of a client.
- *Goal-orientation:* Each company has specific goals set; so does each of the departments. The overall activities of the entity are organized so as to achieve the goals set. If necessary, the goals can be amended to reflect changes in the external environment or other entities/departments.
- *Dynamics and vitality:* The entity adapts to the relevant changes in its external environment and, if necessary, cooperates with entities from other territories—or multiple departments within a single entity cooperate if this is necessary to meet the needs of a client. Moreover, each employee of each department has personal goals defined which allows the entity as a whole to realize individual dynamics.

The company referred to in this example needs to carry out a change management process as it is necessary to implement an IS to support internal risk management procedures. The said procedures are identical across the countries, however more or less stringent restrictions might be imposed in specific countries/departments. Table 2 gives an example of the first step of the change management process. It shows how the basic qualities of fractal system are preserved in the first step of change management process—Identification of the necessary changes. Similarly, following the guidelines expressed in Table 1, enterprise fractality can be preserved in other steps of the change management process. The main issue here is that the fractal qualities are preserved in all the fractals of the enterprise. This issue differentiates this approach from mere adaptive systems which do not take into consideration how changes of one particular subsystem impact the above-mentioned fractal qualities (if such exist) of other subsystems. It shows also that the changes in a fractal system require considerable effort of planning simultaneously at several fractal levels. This differentiates the proposed approach from agile development methods that concentrate mainly on the identified requirements without analyzing the enterprise as a whole [13]. In most cases changes in fractal enterprises involve changes in their IS. In this paper we assumed that each fractal possesses its own IS. This is always the case if IS is considered as a work system and human information processing is included in the definition of the IS [14]. Looking from the point of technical implementation only, one and the same IS or its subsystem (service) can be shared by several fractals. Thus, the fractality of the technical implementation of IS is not

Table 2 An example of change management methodology for fractal systems in the case of implementing a new IS

Basic steps of change management	Change management steps for fractal systems
Identification of the necessary changes	
Identification of necessary changes in strategy, processes, structure, knowledge, information systems in the entire organization or in specific parts thereof	*Self-similarity:* It is necessary to identify whether the changes are needed in the entire organization or in the particular fractal In the current situation it was noted that the information system supporting risk management process has to be implemented in the entire organization. The implemented information system will replace the manual system used differently in every region or even every country. Therefore the change management process will affect the entire company (at country level and department level). By implementing the information system (resource subsystem in Fig. 1, Sect. 2) it will be necessary to change the existing risk management processes (process subsystem), however no major changes will be required in the structure subsystem and goal subsystem *Self-organization:* Two types of changes can be initiated: • In a particular fractal, without changes in the whole system; • In order to achieve better co-operation among the fractals It is necessary to take into account the existing adaptation mechanisms when changes in the entire system are identified. Changes will be made in the company of every country by taking into consideration the co-operation among the companies in different countries and different departments (at country level and at regional level). Besides this—it should be taken into consideration that the main object of the risk management process is the client (company) and the risk level of the same client can be different in different countries. So the information system should support both horizontal and vertical assessment of a client *Goal-orientation:* By identifying changes it should be taken into account that every fractal is goal-oriented and it is possible that changes can impact either the goals of the fractal or the goals of the entire system By implementing an information system supporting the risk management process thus improving the transparency of the process it should be taken into consideration that every employee and every company has strong goal-orientation and implementation of the system can cause disturbances in the existing processes that allow reaching the goals stated by the company *Dynamics and vitality:* By identifying the changes it should be taken into account that fractal systems have a core quality—Coordination and cooperation between self-organizing fractals. Thus it is possible to conclude that changes that are identified in one fractal can theoretically cause changes in other fractals. So it is necessary to identify all the existing relationships among the fractals.

a necessary precondition of a fractal enterprise. Therefore technical issues of IS implementation were not considered in this paper. Nevertheless, it has to be noted that it is quite natural that in fractal enterprises the information system at its technical and work systems abstraction levels might also be organized as fractal system [15, 16].

6 Conclusions and Topics for Further Research

The paper discusses issues relevant to change management in fractal enterprises. It proposes a change management process for fractal systems and shows how it can be applied for IS change management in fractal enterprises. The main focus is on how to preserve the fractal qualities of the enterprise during the change management process, as changes can hinder essential relationships between enterprise subsystems that are behind the self-similarity, self-organization, goal-orientation, and dynamics and vitality of fractal systems. This research only looks at the change management process from an organizational perspective, however further research should also consider the change management process at the level of a technological system as this can have significant influence on the performance of a fractal system.

Depending on the values of change management parameters discussed in Sect 3, different methods can be applied to the change management process. These dependencies and corresponding methods were not considered more closely in this paper; however, they need to be looked at to develop a change management methodology for fractal enterprises which could be used for different types of change management situations in fractal systems of different scales.

References

1. Clarke S (2001) Information systems strategic management. Routledge, London, pp 131–152
2. Warneke HJ (1993) The fractal company. Springer, Berlin
3. Hoverstadt P (2008) The fractal organization: creating sustainable organizations with the viable system model. Wiley, West Sussex
4. Snowdon RA et al (2006) On the architecture and form of flexible process support, software process improvement and practice. Wiley, New York
5. Ryu K, Jung M (2003) Fractal approach to managing intelligent enterprises. In: Gupta JND, Sharma SK (eds) Creating knowledge based organisations, Idea Group Publishers, pp 312–348
6. Cao G, Clarke S, Lehaney B (2003) Diversity management in organizational change: towards a systemic framework. Syst Res Behav Sci 20(3):231–242
7. EBSCO Publishing (Firm) (2007) Bloomsbury Business Library—Business & Management Dictionary. A&C Black Publishers Ltd., London, pp 1473–1473
8. Boddy D, Buchanan D (1992) Expertise of the change agent. Prentice-Hall, London
9. Tharumarajah A, Wells AJ, Nemes L (1998) Comparison of emerging manufacturing concepts. Syst Man Cybern 1:325–331, (Oct)
10. Canavesio MM, Martinez E (2007) Enterprise modelling of a project-oriented fractal company for SMEs networking. Comput Ind. doi:10.1016/j.compind.2007.02.005

11. Ashby WR (1960) An introduction to cybernetics. Chapman and Hall, London, p 295
12. Sprice R, Kirikova M (2006) Feasibility study: new knowledge demands in turbulent business world. In: Nilsson AG, Gustas R, Wojtkowski W, Wojtkowski WG, Wrycza S, Zupancic J (eds) Advances in inforamtion systems development: bridging the gap between academia and industry, vol 2. Springer, New York, pp 131–142. ISBN-10: 0-387-30834-2, ISBN-13: 078-0387-30834-0
13. Stirna J, Kirikova M (2008) How to support agile development projects with enterprise modelling. In: Johannesson P, Soderstrom E (eds) Information systems engineering: from data analysis to process networks, IGI Publishing, Hershey, New York, pp 159–185
14. Alter S (2008) Defining information systems as work systems: implications for the IS field. Eur J Inf Syst 17:448–469
15. Kirikova M (2009) Towards flexible information architecture for fractal information systems. In: Kusiac A, Lee S (eds) The proceedings of the international conference on information, process, and knowledge management, eKNOW 2009. IEEE Computer Society, Washington, pp 135–140
16. Kirikova M (2009) Towards multifractal approach in IS development. In: Barry Ch, Conboy K, Lang M, Wojtkowski G, Wojtkowski W (eds) Information systems development: challenges in practice, theory and education, vol 1. Springer, New York, pp 295–306

Conceptual Modelling for Requirements of Government to Citizen Service Provision

Lily Sun, Cleopa John Mushi and Anas Alsoud

Abstract Government to citizen (G2C) service provision is demanded to fit for purpose for users. A process of finding services from a G2C system involves understanding of the user's request, selecting the relevant services, and deciding amongst the candidate services to meet the user's needs. Some current approaches, such as benchmarking methods, are capable of measuring the service quality in the quantitative manner. However, G2C services also have intangible features which can be measured qualitatively. In this chapter, a method is described to model the requirements of G2C services and their provision. To facilitate a service provision process, a set of criteria is identified and used to ensure the quality of the resultant services. An ontology model developed represents requirements of service provision in a web service environment that involves service consumers, service providers, and service advisor. Interactions between these stakeholders are defined by norms which generate workflows for executing the functions in the techniques. An experiment using DEA is carried out in this paper based on quantitative criteria to validate the method and its techniques, i.e. *articulates, derives* and *pre-selects*. The pre-selected services as the candidates are further evaluated with the contribution of the qualitative features of the services by using analytical hierarchical process (AHP) to decide for winning services.

1 Introduction

Government to citizen (G2C) service provision can be defined as an execution of public services to citizens (i.e. users) that involves various service providers through information and communication technologies (ICT) [2]. As the services become widely online available, the user is also becoming more demanding for fit for purpose services to be provided. This requires an effective approach to benchmark the quality of G2C services and assist in an optimisation process for a right choice of the

C. J. Mushi (✉)
School of Systems Engineering, University of Reading, Whiteknights, Reading, RG6 6AY, UK
e-mail: c.j.s.k.mushi@reading.ac.uk

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_60, © Springer Science+Business Media, LLC 2011

service [10]. However, identifying a suitable service provided by a pool of providers needs an approach to recognise citizens' personalised needs and their needs change. This paper presents an ontology model with a set of techniques that enable a multi-criteria decision making process to provide fit for purpose services to meet citizen requirements in the e-government setting. This model incorporates citizen and service provider profiles which define citizen and service provision requirements. Norms are also built in the model to facilitate a systematic service selection and optimisation process.

The remainder of the paper is organised as follows. Section 2 discusses current trend in measuring quality of G2C service provision and also techniques that can be used in selecting the optimum service requested. Section 3 presents an ontology model with the embedded norm engine which facilitates the requirements analysis for service provision. Section 4 illustrates how the techniques in the ontology model are applied to assist the pre-selection of candidate services and optimise winning services. Section 5 draws conclusion and recommendations for the further work.

2 Related Work

2.1 G2C Service Provision

Currently, web service approach has been adopted by the G2C systems, e.g. DirectGov. Web services support three functions; service description and publishing, service discovery, and service interaction and consumption. Service providers are the ones responsible for publishing their services in the registry from which service consumers can locate and invoke services. The registry (or a public repository) is a public directory containing information about the services and the service providers from diverse sources in a unified way [20]. Web Service driven G2C systems can provide a seamless applications integration that allows the services to be loosely coupled and re-useable across organisational boundaries and from multiple providers via a public repository [18]. However, the current G2C systems are provider-centric by imposing G2C services to be passively provided to users [19]. There are demands for transferring the provider driven approach on to a citizen-centred approach. In a development of citizen driven G2C systems, services should be discovered to meet the user's needs. The traditional web service approach is, to certain extent, able to capture the user's requirements, but the service finding process lacks the capability of comprehensive selection and optimisation of the suitable services. Such capability can not be achieved by keywords matching and smart text mining for similarity, this requires a holistic domain knowledge understanding and representation of G2C service provision. A service register in web services should act as a *service advisor* which applies the underpinning service provision strategies and criteria for service discovery.

The European Commission (EC) defines the G2C services into 12 areas; income taxes, job search, social security benefits, personal documents, car registration, application for building permission, declaration to the police, public libraries, birth and marriage certificates, enrolment in higher education, announcement of moving, and health-related services [5]. It is challenging that a G2C system accomplishes these service provisions via ICT [11], because the dynamic G2C interactivity with complete routine tasks can be sophisticated from both the service providers and users. There is lack of knowledge representation on how to advise quality services offered by the provider to the user. Quality of service in this work is defined through an ontological representation in meeting user needs who expresses a request in the context of higher education service provision. The work extends current e-Government benchmarking approaches by incorporating qualitative criteria in measuring service quality and applying the process of meeting user needs in a web-service environment.

2.2 Service Selection and Ranking Methods

DEA is a linear programming algorithm that has been used successfully in determining the best-practice production frontier relative to individual firms or decision making units (DMUs). In DEA, performance efficiency is measured by a weighted sum of outputs to a weighted sum of inputs for a particular DMU [1, 6]. A DMU represents a service provider who in a pre-selection stage is competitively measured against other providers on their overall performance based on a set of criteria that have been developed. A resultant efficiency, h_o, from DEA is normally decided by either maximizing outputs or minimizing inputs. In our work, maximizing outputs will be considered. DEA is capable of discovering those DMUs which hold a range of value of h_o based on the selection criteria. For those DMUs which satisfy $h_o = 1$, they need to be further decided as a winning DMU for the request. AHP, therefore, is used to assist for such decision making [1, 14].

AHP is a multi-criteria decision making (ranking and optimising) algorithm which enables decision makers who are faced with making numerous and conflicting evaluations to derive a way to come to a compromise [3, 4]. In a set of criteria, each criterion is assigned with significant attribute and each pair of alternatives can be judged by comparing between criteria. Once each comparison has been carried out through all the determining criteria, a winning DMU can be selected.

3 Modeling Requirements of G2C Service Provision

G2C service provision is a knowledge sharing process which requires understanding and representation of the concerns of stakeholders (including users, service providers and service advisor) [8]. An ontological modelling [9] is adopted to define the patterns of behaviour in such a process where various agents with their roles and relationships are involved as well as the norms that govern the service provision workflow.

3.1 *Ontological Modelling for the Requirements of Service Provision*

An ontology model, as shown in Fig. 1, depicts the knowledge of G2C service provision. The model comprises of a set of semantic units, each of which is normally formed by the basic concepts of *agent*, *role*, *affordance*, and *determiner* [7, 13]. An *agent* is drawn with a shape of an ellipse and represents a stakeholder such as an individual (person), a group (organisation) or a community (nation, society). *Affordance* is represented with a rectangle and depicts the relationships of entities and observable pattern of behaviour occurring within a context [9]. An *affordance* forms a semantic unit with a maximum of two antecedents (the concepts on the left end of the line) so as to maintain a concise and stable ontology structure. From the model then, different agents are involved in service provision with various roles such as *citizen, provider, service advisor* and *central government,* which represent responsibilities defined by the corresponding affordances.

All these responsibilities have direct or indirect relations with *products/services* modelled in the ontology. For example, the affordance of *provides* has two antecedents, i.e. *organisation* and *products/services*. The antecedent of *products/services* describes the type of G2C provision and the *organisation* with the role of *provider* provides the G2C services. This semantic unit ensures the service being provided complies with the *public service policies* made by the government. With the affordance of *retrieves*, there are two antecedents of *adopts* and *products/services*. The antecedent of *adopts* defines the quantitative and qualitative criteria based on *public service policies* for a particular *industry* that are adopted by the *organisation* in the role of *service advisor.* The antecedent of *adopts* ensures that there is no conflict of attributes in the *retrieves* process. For example, in universities, sponsorships are available for full-time postgraduate courses only. This therefore resolves conflicts concerning criteria for *course duration, mode of study* and *finance*. The *retrieves* process then contains the focal DMUs with their capabilities in service provision described by the quantitative and qualitative criteria that are defined from adopting the *public service policies*. So, in order to provide a personalised service/product, the model pays special attention to the quality of service provision based on service specification which involves articulation of *requests* of service and citizen's *profile* leading into a pre-selection process with *retrieves*.

The ontology model therefore integrates the requirements for user, service provider, and service advisor with the defined role of *citizen*, *provider* and *service advisor* respectively. Each registered *citizen* has a *profile* which is used to articulate a request for services made to *service advisor*, particularly if the request is fuzzy. For example, if a *citizen* requests for a service of course in Higher Education and this user has some degree of disability. In this case, personal information in the citizen profile will assist to specify the requirements specifications for selecting the service which meets the user's needs. The affordance of *pre-selects* is the one that performs the selection process, and it is ontologically dependant on *articulates* and *retrieves*—the former specifying citizen's request, the latter the DMUs and their capabilities in providing the *services*.

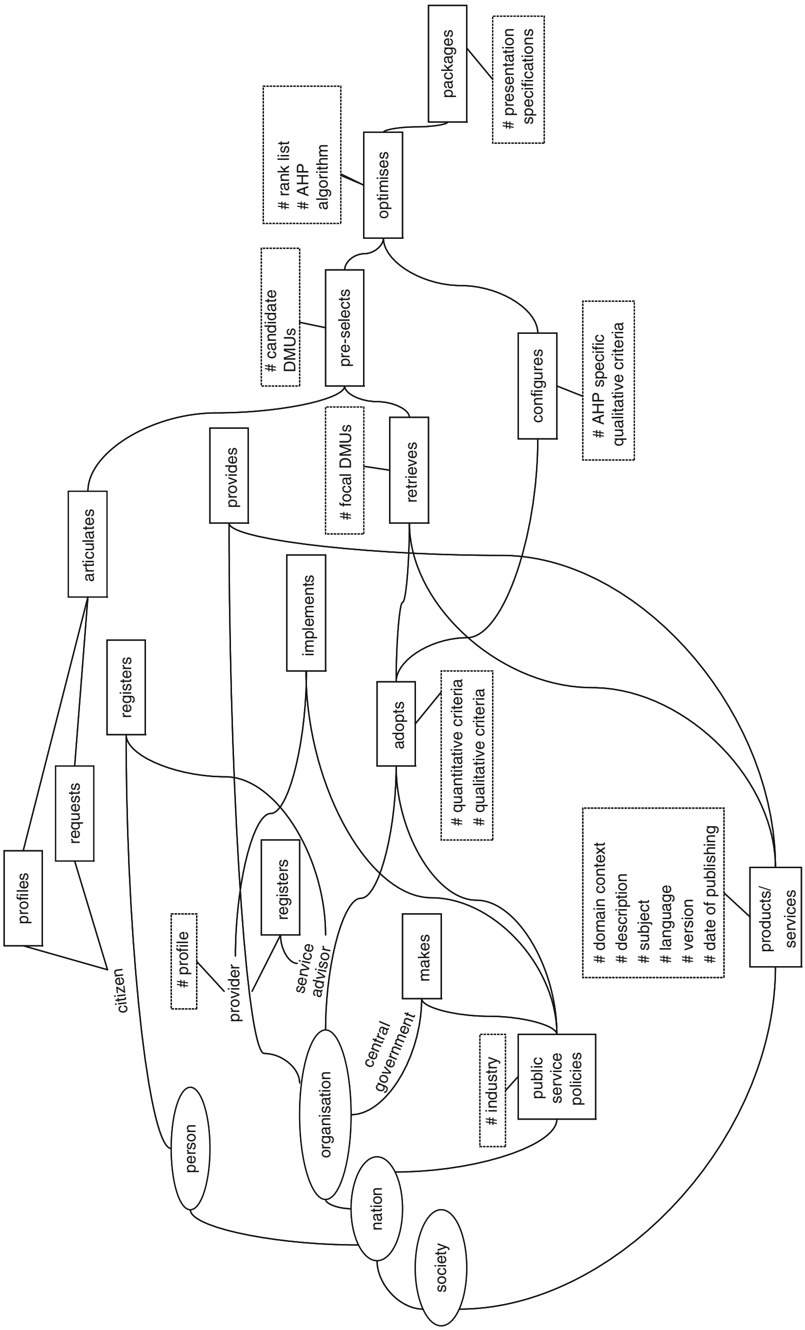


Fig. 1 An ontology model of G2C service provision

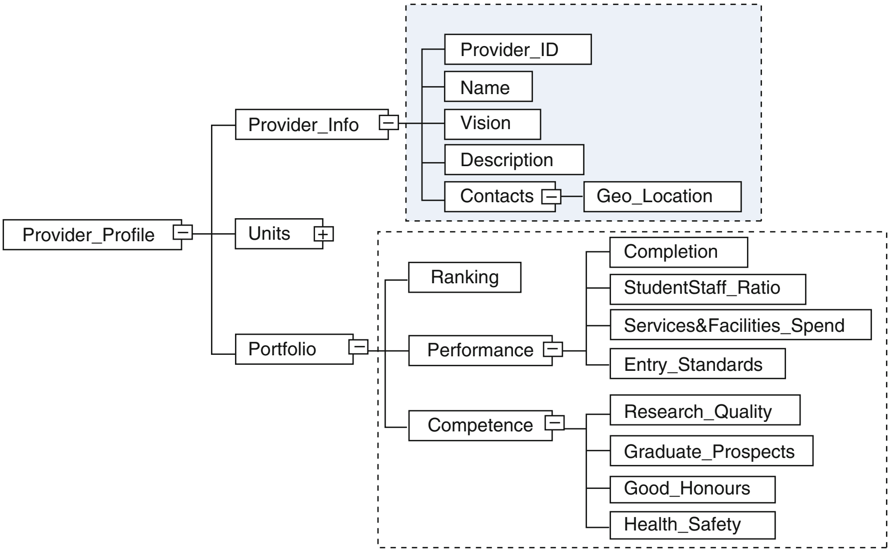


Fig. 2 Service provider profile

3.2 Criteria for Measuring the Quality of Services

The quality of service is largely determined by the service provider who has the reputed capability and competence to offer their services. In order to benchmark service providers for their performance and capability, each registered service provider is described in a provider's profile (see Fig. 2). The *profile* contains the details of a service provider and involves units which provide services. A *Portfolio* describes the capability of the service provision and the quality of the service meeting users' demands. *Provider_profile* assists an identification of focal DMUs while a service provider is being benchmarked in a pre-selection process. This profile also possesses the measures for finding candidate services according to a service request. Table 1 details the service quality measures in the form of criteria.

A service schema in Fig. 3 defines the features and properties of the concept of *products/services* in the ontology model. This schema also captures the criteria in Table 1 for assessing a service. Those service specific criteria can be used to benchmark the quality of the services in conjunction with the provider specific criteria. A set of metadata, such as domain context, description, and language can assist a discovery of the products/services.

3.3 Norms Defining the Service Provision Process

In order to systematically perform a service provision process, norm engine acts as a control mechanism to govern the behaviour of cause of actions. In the ontology

Table 1 Criteria for selecting the relevant *service*

Criteria	Sub-criteria	Type	Description
Quality	Provider ranking	Quantitative	Overall performance
	Subject ranking	Quantitative	Subject performance
	Quality of delivery	Quantitative	Student views on the teaching quality of the subject
	Research assessment	Quantitative	Research quality and innovation
	Entry standards	Quantitative	Requirements for admission of a subject
	Career prospects	Qualitative	Job opportunities and positions for graduates
	Graduate prospects	Quantitative	Destinations of subject graduates
	Student-staff-ratio	Quantitative	Average staffing level for the subject
	Good honours	Quantitative	Percentage of graduates achieving a first or upper second
	Alumni	Qualitative	Experience of graduates
	Accreditation	Qualitative	Reviewed status of institution quality by authorities
	Assessment methods	Qualitative	Ways by which learning outcomes being assessed
	Value added	Quantitative	Student level of achievement from the course
	Feedback	Quantitative	Student judgement of quality of subject they have studied
	Disability services	Qualitative	Means by which various disabilities can be supported
Finance	Academic service spend	Quantitative	Expenditure per student on all academic services
	Facilities spend	Quantitative	Expenditure per student on staff and student facilities
	Tuition	Quantitative	Monetary value to studying the program per academic year
	Financial support	Qualitative	Bursary, sponsorships and funding for the course
Flexibility	Mode of study	Quantitative	Full-time/part-time/distance learning
	Academic calendar	Quantitative	Different starting dates of the course
Geog. location	Country	Quantitative	The country where the service is offered

model, each affordance of the semantic unit is associated with a period of existence which can be defined by a pair of *startNorm* and *finishNorm*. During the existence, *operationalNorm* describes its pattern of behaviour. For example, the *startNorm* and *finishNorm* define the *pre-selects* existence and its cease (see Fig. 4). The *operationalNorm* specifies the DEA algorithm that takes multiple input and multiple output quantitative criteria for selecting all competitive DMUs based on their maximum efficiencies, i.e. $h_o = 1$. The operational norms are normally defined for generating a

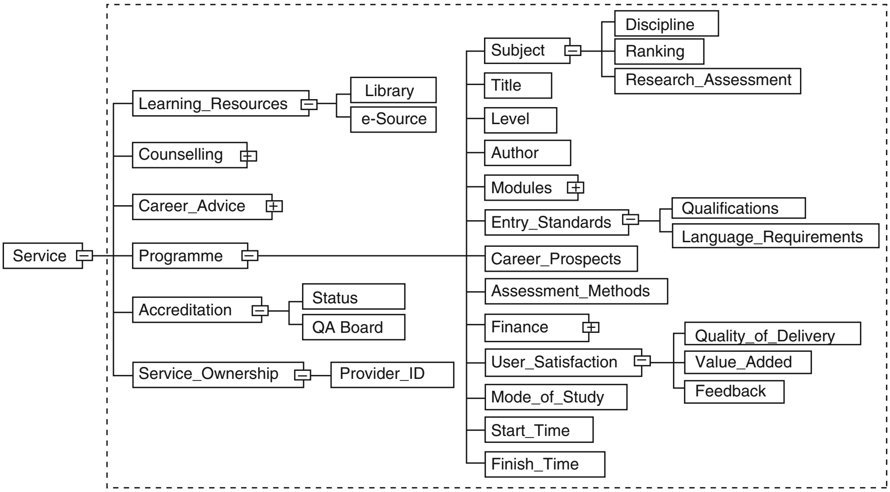


Fig. 3 Service schema

<?xml version="1.0" encoding="UTF-8"?>
<Norm_pre-selects>
<StartNorm ID=N1.1>
<whenever> a citizen requests a service via a service advisor </whenever>
<if_condition> the services AND the quantitative criteria can be retrieved</if_condition>
<then_agent> pre-selects engine </then_agent>
<is_deontic_operator> permitted </is_deontic_operator>
<to_action> execute the pre-selection workflow </to_action>
</StartNorm>
<finishNorm ID=N1.2>
<whenever> a citizen requests a service via a service advisor </whenever>
<if_condition> the services AND the existing quantitative criteria can no longer be found </if_condition>
<then_agent> pre-selects engine </then_agent>
<is_deontic_operator> obliged </is_deontic_operator>
<to_action> cease the pre-selection workflow </to_action>
</finishNorm>
<operationalNorm ID=N1.3>
<whenever> a citizen requests a service via a service advisor </whenever>
<if_condition> the service specific & provider specific criteria are available AND the request is clearly articulated </if_condition>
<then_agent> pre-selects engine </then_agent>
<is_deontic_operator> obliged </is_deontic_operator>
<to_action> select AND rank all the DMUs whose h_0 =1 by using DEA algorithm:

$$Max \quad h_0=\sum_r u_r y_{rj_0} \;\Big|\; \sum_i v_i x_{ij_0}=1 \qquad \sum_r u_r y_{rj} - \sum_i v_i x_{ij} \le 0 \;\big| j=1,2,\ldots,n \big|\; u_r, v_i \ge 0 \big|$$

</to_action>
</operationalNorm>
</Norm_pre-selects>

Fig. 4 The norms defining the *pre-selects* process

workflow of each affordance in the semantic unit. The result from the left semantic unit is carried over as the input to the workflow in the right semantic unit. All the workflows can be integrated to structure the entire service provision process.

4 Application and Discussion of the Ontology Model

The ontology model and the techniques are validated through various simulations towards the selected service requests. A service selection process normally starts from a service request. For example, a *person* with the role of *citizen* requesting for a higher education programme, e.g. a master course in Computer Science from a top 10 university in the UK that starts from October and also offers studentships. Based on the request, the *articulates* deduces a set of multiple criteria of {*Geo_Location, Ranking, Graduate_Prospects, Entry_Standards, Good_Honours, Completion*} from the provider's profile as well as {*Title, Ranking, Entry_Standards, Career_Prospects, Graduate_Prospects, Financial_Support, Level, Start_Time, Quality_of_Delivery, Value_Added, Feedback*} from the service schema. Each of these criteria is assigned with a weight which is determined by the requestor's preference in conjunction with the pre-defined domain rules. The *pre-selects* then uses the criteria to discover the DMUs. There are two sets of results produced to represent the quality of the service on the basis of institutional and subject level respectively (see Tables 2 and 3).

The results in Table 2 show that only INS-222 and INS-847 satisfy the maximum efficiency, i.e. $h_o = 1$ and also fall into top 10 institutional ranking [16, 17] that primarily meet the user's request. The *working* column shows the difference between weighted output (W/O) and weighted input (W/I) as constraint derived from $\sum_r u_r y_{rj} - \sum_i v_i x_{ij} \leq 0$, where $j = 1,2,\ldots,n$ and $u_r, v_i \geq 0$. The results in Table 3 present the subject quality measures independently from the general performance of the institutions. It is observed that a subject ranking position many not be the same as its provider's. By comparing these two sets of results, it is expected to identify a discrepancy, particularly when there is no link between them at all.

Table 2 The consolidated results for the DMUs based on provider rankings

DMU	Input		Output					
	Provider ranking	Entry standards	Completion	Good honours	W/O	W/I	Efficiency	Working
INS-037	5	0.4732	1.0000	0.7990	0.9699	1.0000	0.9699	−0.0301
INS-085	3	0.5045	1.0000	0.7062	0.8960	1.0000	0.8960	−0.1040
INS-118	9	0.4588	1.0000	0.7823	0.9871	1.0000	0.9871	−0.0129
INS-133	1	0.5346	1.0000	0.9296	0.9561	1.0000	0.9561	−0.0439
INS-222	7	0.4978	1.0000	0.9053	1.0000	1.0000	1.0000	0.0000
INS-366	4	0.5031	1.0000	0.7917	0.9069	1.0000	0.9069	−0.0931
INS-424	2	0.5439	1.0000	0.8788	0.8973	1.0000	0.8973	−0.1027
INS-522	6	0.4823	1.0000	0.8302	0.9752	1.0000	0.9752	−0.0248
INS-847	10	0.4521	1.0000	0.7802	1.0000	1.0000	1.0000	0.0000
INS-975	8	0.4914	1.0000	0.8739	0.9896	1.0000	0.9896	−0.0104

Table 3 The consolidated results for the DMUs based on subject rankings

DMU	Subject ranking	Provider ranking	Entry standards	Quality of delivery	Value added	Feedback	Graduate prospects	Efficiency
INS-037	31	5	0.4614	0.7500	0.0341	0.4659	1.0000	1.0000
INS-085	7	3	0.4915	0.8723	0.0319	0.6277	1.0000	1.0000
INS-118	18	9	0.5256	0.9634	0.0610	0.7073	1.0000	0.9838
INS-133	1	1	0.9566	0.9566	0.0361	0.6831	1.0000	0.9222
INS-222	2	7	0.4915	0.9314	0.0704	0.6651	1.0000	1.0000
INS-424	3	2	0.6359	0.9565	0.0217	0.8152	1.0000	1.0000
INS-522	5	6	0.5554	1.0000	0.0482	0.7952	1.0000	1.0000
INS-847	8	10	0.4943	1.0000	0.0455	0.5114	0.9091	0.8754
INS-975	26	8	0.5063	0.8750	0.0500	0.6625	1.0000	1.0000

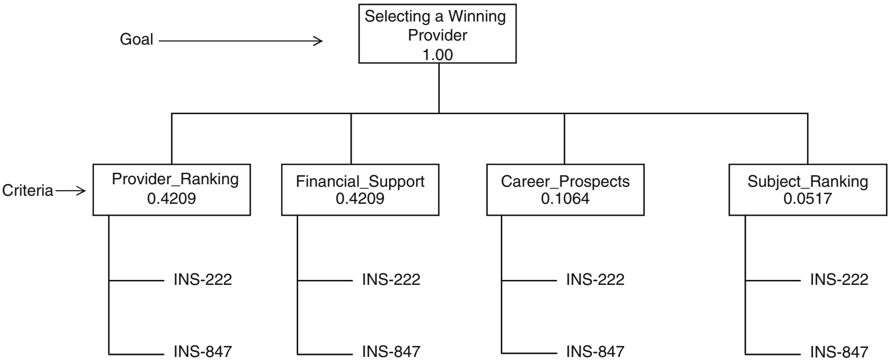


Fig. 5 The decision tree with the criteria weights

Based on the subject ranking, six out of nine institutions satisfy the maximum efficiency with INS-222 achieving the maximum efficiency in both provider and subject rankings. The two sets of results suggest: (1) getting an optimum result based on the subject of interest from the pre-selected institutions, i.e. INS-037, INS-085, INS-222, INS-424, INS-522 and INS-975; (2) considering the institution which has been pre-selected in both cases, i.e. INS-222; or (3) getting an optimum result based on the pre-selected institutional rankings. To follow the third suggestion which is the original user request, the list of the candidate DMUs produced by *pre-selects* is further analysed for optimisation. The affordance of *optimises* determines the winning service by using the AHP algorithm with specific qualitative criteria, i.e. *Financial_Support, Career_Prospects, Provider_Ranking* and *Subject_Ranking.* Figure 5 represents the decision tree for multi-criteria evaluation.

In order to ensure a consistency in prioritisation, consistence ratio $CR = (CI/RI)$ is computed. *CI* represents consistency index, $\lambda_{max} - (n/n)$, where λ_{max} is the principal eigen value, *n* is the number of pair-wise comparison matrix, and *RI* is Random consistency Index. In this simulation, $CR = 2.72\%$ shows that the preferences for 4 criteria are consistently evaluated, i.e. $CR \leq 10\%$ [12, 15]. Due to the user request, *Provider_Ranking* and *Financial_Support* are assigned more weights.

Table 4 Overall composite weights of the alternatives

	Provider_ranking	Financial_support	Career_prospects	Composite_weight	Rank
Adjusted weights	0.4439	0.4439	0.1122	1.0000	
INS-222	0.8333	0.2500	0.5000	0.5370	1
INS-847	0.1667	0.7500	0.5000	0.4630	2

AHP computation shows that *Subject_Ranking* has very little significance (i.e. 0.0517) compared to the others. Therefore, it was ignored in the further computation. Consequently, the weights for the other three criteria have to be re-adjusted, such as 0.4439 for *Provider_Ranking,* 0.4439 for *Financial_Support,* and 0.1122 for *Career_Prospects*. Table 4 consolidates the overall composite weights of the two DMUs and INS-222 satisfies the user's request.

5 Conclusion and Future Work

G2C service provision presents a complex problem domain that demands a method to model requirements of service provision. This research work has proposed the use of ontology model for conceptualising the domain knowledge where citizen requirements are presented from the aspect of social domain. The ontology model presents very well the concept of web service with different stakeholders such as *service advisor, central government,* and *service providers* interacting to ensure that quality of G2C service provision is maintained. Profiles were also created to facilitate articulation of *requests* and to support for *personalised* service provision. Service schema was developed to illustrate different attributes and criteria for the service provided. Norms were embedded in the model for the execution of a pre-selection workflow. DEA has been applied to implement *pre-selects* process due to the multi-criteria nature of the problem domain which also exhibits multiple output and multiple input service criteria. The results indicate that DEA has been effective in selecting suitable services that can meet citizen request.

The model also provides for alternative suggestions to the user to capture intentions and what can also be provided in line with the user request which the user might not be aware. This forms the intelligent part of the system which calls for further work. To process further the request from pre-selection, an evaluation was conducted to compare the differences between the needed qualitative criteria of the selected service providers. AHP was used as a mechanism to prioritise the selected services based on weighting and the optimum service candidate capable of meeting citizen requirements was identified. Future works will also need to identify mechanism to assist the selection in the problem domain through matching cases, and to develop a CASE tool to facilitate consistent documentation.

References

1. Azadeh A, Ghaderi SF, Izadbakhsh H (2008) Integration of DEA and AHP with computer simulation of railway systems improvement and optimization. J Appl Math Comput 195:775–785
2. Bekkers V, Homburg V (2005) E-government as an information ecology: backgrounds and concepts. In: Bekkers V, Homburg V (eds) E-government as institutional and technological innovation in public administration. IOS Press, Amsterdam, pp 1–20
3. Benyoucef M, Canbolat M (2007) AHP-based supplier selection in e-procurement. Int J Serv Oper Manag 3:172–192
4. Chen CT, Lin CT, Huang SF (2006) A fuzzy approach for supplier evaluation and selection in supply chain management. Int J Prod Econ 102:289–301
5. Graafland-Essers I, Ettedgui E (2003) Benchmarking e-government in Europe and the US. Statistical Indicators Benchmarking the Information Society, Bonn, Germany
6. Hasan AM, Shankar R, Sarkis J (2008) Supplier selection in an agile manufacturing environment using data envelopment analysis and analytical network process. Int J Logist Syst Manage 4:523–550
7. Liu K (2000) Semiotics in information systems engineering. Cambridge University Press, Cambridge
8. Liu K, Clarke RJ, Andersen PB, Stamper, RK (2001) Information, organisation and technology: studies in organisational semiotics. Kluwer, Boston
9. Liu K, Sun L, Fu Y (2008) Ontological modelling of content management and provision. J Inf oftw Technol 50:1155–1164
10. von Lucke J, Reinemann H (2000) Speyerer definition von electronic government. http://foev.dhv-speyer.de/ruvii/Sp-EGov.pdf. Accessed 25 Mar 2010
11. Ronagham SA (2002) Benchmarking e-government: a global perspective. United Nations Division for Public Economics and Public Administration, New York
12. Saaty TL (1990) Eigenvector and logarithmic least squares. Eur J Oper Res 48:156–160
13. Sun L, Ousmanou K, Cross M (2008) An ontological modelling of user requirements for personalised information provision. J Inf Syst Front 12:337–357
14. Sung HH, Krishnan R (2008) A hybrid approach to supplier selection for the maintenance of a competitive supply chain. J Exp Syst Appl 34:1303–1311
15. Tan PS, Lee EW, Mous K, Lee SSG, Goh AES (2008) Multi-criteria, context-enabled B2B partner selection. IEEE international conference on systems, man and cybernetics, Singapore. doi:10.1109/ICSMC.2008.4811533
16. The Complete University Guide (2010) University league tables 2010. http://www.thecompleteuniversityguide.co.uk/. Accessed 4 Mar 2010
17. The Sunday Times University Guide (2010) University rankings league tables. http://extras.timesonline.co.uk/tol_gug/. Accessed 4 Mar 2010
18. van Thanh D, Jorstad I (2005) A service-oriented architecture framework for mobile services. In: Proceedings of the advanced industrial conference on telecommunications/service assurance with partial and intermittent resources conference/e-learning on telecommunications workshop (AICT/SAPIR/ELETE'05). IEEE Computer Society, Washington, pp 65–70
19. Wang H, Huang JZ, Qu Y, Xie J (2004) Web services: problems and future directions. J Web Semant 1:309–320
20. Zou G, Xiang Y, Gan Y, Wang D, Liu Z (2009) An agent-based web service selection and ranking framework with QoS. In: 2nd IEEE international conference on computer science and information technology. doi:http://doi.ieeecomputersociety.org/10.119/ICCSIT.2009.5234607

On the Broader View of the Open Source Paradigm: The Case for Agricultural Biotechnology

Meredith A. Taylor, Geoffrey Black and Wita Wojtkowski

Abstract This study reviews the use and advantages of open source software licensing and examines how the use of similar arrangements could increase productivity and innovation in other industries where intellectual property rights (IPR) protection has relied on proprietary measures and where rapid technological innovation is important. We observe that a reliance on proprietary, closed source protection of intellectual property in agricultural biotechnology has hindered product development and resulted in spiraling research and development costs. Reduced research productivity and increasing costs are of serious concern to practitioners, academicians, and policy makers. We show that lessons learned from the open source initiatives in software technology are directly applicable to agricultural biotechnology.

Abbreviations

BIOS	Biological Innovation for Open Source
CAMBRIA	Center for the Application of Molecular Biology to Industrial Agriculture
EPA	Environmental Protection Agency
EU	European Union
FDA	Food and Drug Administration (US)
FSF	Free Software Foundation
GM	Genetically Modified
GPL	General Public License
IPRs	Intellectual Property Right
OSI	Open Source Initiative
TRIPs	Trade-Related Aspects of Intellectual Property Rights
USDA	United States Department of Agriculture

M. A. Taylor (✉)
University of Bern, 6 Hallerstrasse, Bern, Switzerland

Boise State University, 1910 University Drive, Boise, Idaho, USA
e-mail: meredithtaylor@boisestate.edu

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_61,

USTR	United States Trade Representative
WIPO	World Intellectual Property Organization
WTO	World Trade Organization

1 Introduction

The process of open source innovation and production is dramatically different than the ways in which commercial products are traditionally produced with private firms employing workers who receive compensation for developing new products and with retaining private property rights. In the case of intellectual property, the compensation received from selling usage rights provides the incentive for firms to develop new products and, in turn, to compensate workers for product innovation. Firms can protect their rights to intellectual property by the use of measures such as patents and copyrights. Such a legal framework allows firms to generate revenues from the sale of *proprietary* products in order to recoup research and development and other commercialization costs. Further, *proprietary rights* are seen as providing incentives for innovation and new product development (*see*, for example, Lerner and Tirole 2005; Goldman and Gabriel 2005).

In contrast to this traditional model, based on proprietary product development, an open source regime generally involves developers of intellectual property making content publicly available without charge. In open source environments, management is limited, workers are frequently unpaid, legal restrictions on the use, alteration, or redistribution of the product is meager. Further, product development often proceeds via peer-production among loosely collaborating individuals often spread around the world and without financial links. As a result, economists note that the incentive structure in open source environments is fundamentally different from that in proprietary environments (*see*, for example, Maurer and Scotchmer 2006; Lerner and Tirole 2000, 2005). However, these alternative incentives are functioning and effective is evident from the fact that open source development is increasingly operative in industries where innovation and the rapid development of intellectual property are crucial (Gurbani et al. 2006; Munga et al. 2009; Nagy et al. 2010).

Open source approaches are particularly well established in the development of computer software. In economic terms, open source software is similar to privately-produced public goods in that it is publicly available and redistributable (O'Mahoney 2003; Baldwin 2008). Being publicly available does not, however, mean that open source software is similar to material in the public domain, in which material receives no protection under intellectual property law systems (*see* Guibault and Hugenholtz (2006) and Samuelson (2006) for extensive treatments on public domain law). In contrast to public domain material, open source software is subject to a variety of licensing arrangements. While these are generally much less restrictive than patent or copyright protection in the proprietary environment, open source software developers are able to retain rights to limit the use and redistribution of their work.

There are advantages to the use of open source licensing arrangements that are evident in the development of computer software. This study reviews the use and advantages of open source software licensing and examines how the use of similar arrangements could increase productivity and innovation in other industries where intellectual property rights (IPRs) protection has relied on proprietary measures and where rapid technological innovation is becoming increasingly important.

This study begins with a brief review of the development of the open source environment in the software field. Next, we review the IPR protection regime in the field of agricultural biotechnology and the reliance on traditional proprietary measures both domestically and internationally. Finally, we posit that reform of domestic and international IPR protection through the adoption of more open source methods may yield gains for innovation and food production, generally.

2 The Open Source Paradigm in Software Development

The open source environment provides a constructive framework for research, knowledge production, and product development. The open source model began as an alternative approach to software development in the early 1980s and, as a result, is well known and well developed for computer software. In the open source software arena, programs are made publicly available for use, alteration, or dissemination under relatively unrestrictive licensing agreements.

The history of the open source software movement entails a shift from a closed source approach, in which alteration of source code is not allowed, to an open source approach. One of the first efforts to formalize the licensure of open source software was the formation of the Free Software Foundation (FSF) by Richard Stallman in 1983 and the development of the GNU project and the accompanying General Public License (GPL) licensing agreement. GPL licensees allow these four actions: the (1) freedom to run the software, for any purpose, (2) freedom to study how the software works, and adapt it to his/her needs, (3) freedom to improve the software, and release his/her improvements to the public, so that the whole community benefits, and (4) freedom to access to the source code (Free Software Foundation 2009). The creation of the Open Source Initiative (OSI) by Eric Raymond and Bruce Perens in 1998 established an administrative body that, along with FSF, provides for the certification of open source licenses (Raymond 1999). In general, the distribution of open source licensed software must allow free redistribution, must include source code, must allow modifications and derived works, must not be tied to a particular software distribution or technology, must not restrict other software, and must not discriminate against persons, groups, or fields of endeavor.[1]

Approved licenses are recorded and administered through the OSI and the FSF.[2] In general, open source licenses carry few restrictions and facilitate collaboration

[1] *See* Open Source Initiative (2010a), http://www.opensource.org/docs/osd

[2] For a list of these licenses, *see* Open Source Initiative (2010b), http://www.opensource.org/licenses

by allowing developers to use, copy, change, improve, and redistribute software. Moreover, the relative paucity of restrictions in open source licensing and resulting collaboration among developers mean that an accumulation of knowledge is facilitated through motivating contributions within a cooperative development effort by people with similar interests in product improvement. Most importantly, spontaneous formation of collaborative development groups leads to a peer-production process that creates new and improved technologies.

In the development of open source software, one of the best-known examples of a software development community is the ongoing development of the Linux operating system originally developed by Linus Torvalds as a free UNIX kernel. Torvalds utilized the FSF to create a free UNIX kernel that was transformed into a complete and full-featured distribution of the UNIX operating system to be freely distributed with the source code. The result is a competitive and stable system that runs millions of servers and is found in a host of applications from cell phones to supercomputers.

In the development of UNIX and other open source software, it was found that a critical part of the process of innovation is access to previously developed code (Hope 2008). Just like inventors who build upon the inventions before them, software innovators need access to current technological building blocks in order to release new architectures. Developers found that, compared to the traditional closed source model, the open source model allowed for a more rapid process of software evolution. It is important to note that open source development does not mean that the resulting product is strictly non-commercial. According to the FSF, "a free program must be available for commercial use, commercial development, and commercial distribution."[3] Most significantly, developers benefit from implementing an open source model because they are then able to gain access to ideas and input from a greater variety of sources, beyond their own businesses.

We posit that in agricultural biotechnology there are similar economic and sociological incentives and merits to having basic scientific research conducted through the utilization of such collaborative approaches. We develop our argument in the next sections of this paper.

3 Intellectual Property Rights and Innovation in Agricultural Biotechnology

Intellectual property rights may either greatly promote or hinder innovation and knowledge development. The formalized construction of a process designed to privatize intellectual accomplishment in biotechnologies has again forced innovators, organizations, negotiators and scholars into new debates over what, how and to what extent IPRs can and should be protected in light of the numerous public interest needs involved. Specifically, current regulatory and policy reform has been occurring in

---

[3] *See* GNU Project—Free Software Foundation (2009), http://www.gnu.org/philosophy/free-sw.html

response to special developments and challenges in rapidly evolving industries. As a result, IPR protection regimes are evolving differentially across industries. The open source paradigm is recognized as a means of ensuring interaction and cooperation among independent participants and, given that the free exchange of information can be unduly restricted by IPR protection regimes, is being adopted increasingly across industries. While it is currently widely utilized in industries such as computer software, it is only being slowly embraced in others, including biotechnology.

Designing and implementing domestic and international regulations and policies aimed at intellectual property protection proves especially challenging in the area of agricultural biotechnology. Genetic modification technology plays a significant role in food growth, safety and production. Agricultural productivity is an essential component for sufficient food production, globally, and is especially critical for developing countries. The commercialization of agricultural biotechnology research and development has evolved as investment trends have shifted toward this industry, but has maintained, to a large degree, a reliance on proprietary rather than open source, IPR protection. Tantamount to the following discussion is whether the scope of protection given to the holders of IPRs is too broad, creating a deterrent for scientists to engage in open communication with one another, and thereby hindering further innovation and for food production in the long term.

Profound recent changes in the technological advances in the field, as well as in the constantly evolving underlying protection regime, have been fraught with controversy. The biotechnology boom has required legal and legislative action to define, monitor and regulate the biotechnology industry. However, the results have not all been positive: public support for scientific research has declined; venture capital and other private funding sources are not as readily available as before; the role of academic institutions to bridge the gap between industry and university has become muddled as the academic and commercial spheres have collided and *seemingly* merged (Hope 2008). Further, there have been concerns over the integrity of having a research process being fundamentally based on a commercial model.

The United States' policy on intellectual property rights for life sciences generally provides for broad patent rights to private industries. An environment amenable for the privatization and rapid commercialization of biotechnology research has resulted from the continuation of policies adopted early in the regulatory arena and supported by case law. Initial steps in the regulatory arena began with the 1980 Patent and Trademark Law Amendments Act (P.L. 96-517 or "Bayh-Dole Act"[4]) which significantly furthered the privatization and commercialization of biotechnology research by reversing the presumption that any governmentally-funded research would be considered and later treated as a public resource (Hope 2008). The effects of this act were strengthened by the Office of Science and Technology, an executive agency, with the 1983 establishment of the "Coordinated Framework for the Regulation of Biotechnology." This Framework sets out, for example, how the development of transgenic plants would be regulated and commercialized through three regulatory

[4] Effective July 1, 1981. Later amended by PL 98-620 on Nov. 8, 1984; and consolidated and finalized through 37 CRF Part 401 (52 FR 9552, March 18, 1987).

bodies, the Food and Drug Administration (FDA), the Environmental Protection Agency (EPA), and the United States Department of Agriculture (USDA).[5] The impact of the passage of these and similar legislation has increased significantly over time as the effects of the patenting regimes were realized and strengthened by case law. In this light, one of the most significant of these is the *Diamond v. Chakrabarty* case, which played a vital role in transforming the patentability of living organisms.[6] The majority ruled that "anything under the sun that is made by man" was patentable, whether it was a living or an inanimate object.[7] As the biotechnology industry evolved over time, the effects of such regulatory actions and legal findings have led to increased concern that the reliance on privatization and commercialization within the agricultural biotechnology industry constitutes a threat to scientific progress by tipping the balance of incentives away from contribution to a common fund of knowledge and towards restrictive communication practices motivated by the prospect of private ownership of scientific knowledge (Hope 2008).

The emphasis on the privatization of intellectual property has also informed international regulations and policies protecting intellectual property rights. These have evolved rapidly over the past three decades in response to changing economic demands and the technological advances, which they seek to protect and for which they seek to give allowances within the world trading system. More specifically, these regulations and policies have designed a means by which traditional IPRs, such as patents, copyrights and trademarks, are created and enforced in the global trading arena.

Private industries have united on a national level to have their interests championed internationally. In the United States, this is primarily the responsibility of the Office of the United States Trade Representative (USTR). At the global level, landmark adoption of the Trade-Related Aspects of Intellectual Property Rights (TRIPS) agreement, during the Uruguay Round of multilateral trade negotiations in 1994 by today's members of the World Trade Organization (WTO), is the controlling international agreement on patent policy, including plant and seed patents. The TRIPS agreement sets forth the minimum standards of protection that the great majority of economies of the world must currently recognize and respect. This hallmark development was subsequently incorporated through the adoption of other IPRs rules in bilateral and regional trade agreements, and intergovernmental treaties—most commonly found under the direction and auspices of the World Intellectual Property Organization (WIPO). The future path of biotechnological development and, to some extent, the legitimacy of the WTO depends on how, for example, genetically modified (GM) crops are treated and implemented in domestic as well as international regulations (Winickoff et al. 2005). Although one of the main objectives of

[5] The FDA determines whether genetically modified organisms are "safe to eat"; the EPA determines whether they are "safe for the environment"; and the USDA determines which foods are "safe to grow" (Wong 2003; Grossman 2001).

[6] *Diamond v. Chakrabarty*, 447 U.S. 303,309 (1980). Prior to this time, the US Patent Office would refuse applications for patents on living organisms.

[7] *Diamond v. Chakrabarty*, 447 U.S. 303,309 (1980) quoting SEN.REP.No.1979 (1952).

IPRs policy is to create additional incentives for individuals and firms to engage and invest in making new discoveries and developing new solutions to problems faced by the world, concerns have been raised that ongoing or future innovation will be hindered due to an inability to access and utilize protected intellectual property.

4 Open Source in Agricultural Biotechnology

There is a strong parallel between open source software licensing and agricultural biotechnology licensing. The great similarity is their dependency on bioinformatics, or the use of coding for desired traits or commands. In the case of biotechnology, this requires access to genetic sequences and functional information (Winickoff et al. 2005). There have been a number of agricultural biotechnology practices that are now altered due to the privatization and commercialization of agricultural biotechnology. Most generally, while research has shifted from the public to private sphere (Wang et al. 2009), there exists a need for access to and understanding of the information from earlier inventions, on which new inventions would be created.

Agricultural biotechnology pulls from a number of academic disciplines and requires an accumulation and coordination of knowledge. Each invention has been built upon the last and requires input from multiple fields. At present, new hybrids of intellectual property protection beyond patents and trademarks have been created including: geographical indicators, trade secrets and contracts, and plant breeders' rights.[8] These developments indicate a greater proliferation and strengthening of intellectual property rights in this industry, which further detriments the opportunity or creation of a free exchange of information and ideas. With the increased use of potential proprietary inhibitors hindering exchange between researchers in these varying disciplines, innovation has arguably mired.

There are several alternative approaches that could adopt aspects of the open source model. In general, open source agricultural biotechnology projects would utilize the licensing scheme from the patent system but, at the same time, guarantee that the critical underlying technologies of an innovation remain available for utilization by other inventors. In this approach, when an inventor obtains an open source license, s/he is then permitted access to the requisite information to then make an improvement to the invention—something that was not previously part of the original patent. The original patent would later be credited, but the owner would not require additional royalties from the new inventor. Other solutions have been proposed to be implemented through the traditional circuits of change: law reform; an increase in public funding; reduction in transaction costs. Further, there has been support for the creation of collective rights organizations to assist in creating collaborative environments.

[8] In the US, these are called Plant Variety Protection Certificates.

4.1 Select Examples

There are a number of examples of how open source has proven beneficial to the area of agricultural biotechnology. One is TransBacter, a method of gene transfer for plants using bacterial species designed to be a work-around to the many patents covering bacterial genes and thus aims to overcome the current IPR restrictions to the commercialization of products created using bacteria-mediated gene transformation in plants (Broothaerts et al. 2005). Another example is the Tropical Disease Initiative, a web-based environment where scientists from universities, laboratories, and corporations work together on early-stage development of drugs to fight tropical diseases via a shared database (Ortí et al. 2009). A similar partnership of scientists, the HapMap Project, aims to develop a specialized map of the human genome. The goal of the project is to develop a map that describes not the standard sequence of human DNA but rather common patterns of human DNA variation, in order to determine which genes are associated with certain diseases (International HapMap Consortium 2007; Feldman 2004). The consortium promotes the free exchange and access of information generated by the project.

Perhaps the best and most established example of such collaborative open source-based groups is the Biological Innovation for Open Society (BIOS) project, based on the protected commons approach (Jefferson 2008). In this approach, a group of technologies sufficiently effective to assist in basic research and useful for small biotechnology companies or developing countries (a BIOS "node"), is licensed in a way that allows for cooperative innovation and technological collaboration. Other initiatives, such as the Center for the Application of Molecular Biology to International Agriculture (CAMBIA), have made such projects possible. Through the establishment of cooperative ventures between collective open-source and protected commons model projects, core technology is available for further progressive innovation. The BIOS License[9] permits users to patent any inventions they create as a result of the collaboration. The key to the project is that the core technologies and tools are released as public goods. Thus, the innovation in biotechnology, as well as the public interest, is greatly enhanced through the promotion of innovative legal instruments such as the BIOS License.

4.2 The Case for Caution

While there are legitimate justifications for defending the need for enhancing and preserving cumulative innovation via the use of open source models, their wholesale adoption in the agricultural biotechnology industry should be considered with

[9] The license makes the content free of charge to anyone. Users are required to sign a license agreeing to grant back any improvements in the core technology and to make such improvements freely available to all other users on the same terms that BIOS provided for the original core technology.

caution. First, there are a number of economic justifications for having intellectual property protection. Primarily, intellectual property protection has been implemented as a means of motivating and stimulating technological innovation. Economists argue that, by allowing an inventor to have an exclusive right to commercially exploit their invention for a limited time, a necessary incentive is provided for them to decide to engage in the process of innovation (*see*, for example, Harrison and Koski 2009). However, this approach creates expectations and assumes motivations based solely on monetary incentive schemes. Further, due to the exclusive nature of the patenting system, consumer prices are higher than in competitive markets because of the monopoly power granted to the inventor. As a result, while incentivizing innovation, patents tend to impede other firms from freely adopting, imitating or improving patented ideas without the original patentee's consent as well as in market-power pricing.

A second area of concern for the use of open source agricultural biotechnology involves the applicability and transferability of licensing and development principles, learned and experienced during the free software movement, to the agricultural biotechnology industry. There are a number of factors, both internal and external, that affect the efficacy of the use of open source software models for biotechnology. Internally, structural differences between the computer software and biotechnology industries are causes for concern. These include disproportionate capital costs of development, industry cultures, and varying applicable legal regimes. In addition, external political, economic, and social factors have caused policy reforms to be considered with hesitation and uncertainty. Consider that in the development of new biotechnological methods, organisms, and food products, only a few individuals have access to full information and, consequently, an understanding of the innovations taking place. This has had important impacts in the field of genetically modified (GM) food development, causing a lack of support for GM foods.

This is significant in the policy arena because the failure to garner public support has been one of the impediments to the adoption of GM foods both domestically and internationally. Uncertainty in this area will likely influence the future of the global trading regime of the WTO, specifically its ability to address global concerns as far-reaching as human rights, health, and its primary objective of reducing trade barriers and protectionist practices. An important example is the WTO dispute settlement procedures between North America (and Argentina) and the European Union (EU), based on a controversy over the EU's *de facto* ban or non-approval of GM crops. It has been argued that there is a need for greater scientific justification and risk assessment throughout international trading practices. One proposed solution is to increase public participation so as to capture cultural value differences. Another is the creation of an integrated system comprising scientific and legal regulatory experts during the dispute settlement proceedings to assess and define risk and that taking a comprehensive approach may lead to "objective assessment of the facts" (Winickoff et al. 2005). This standard could then be applied during the proceedings to determine whether member states took legitimate steps toward assessing risk.

On the domestic level, numerous attempts have been made in the professional, academic, and policy arenas to increase public awareness and understanding. One example is the Pew Initiative on Food and Biotechnology, established with the aim of providing an independent, non-biased, and objective source of information on agricultural biotechnology for use by the public, media and policymakers. In addition, this initiative sought to collect and distribute information and encourage debate to help individuals and governments to making informed decisions.[10]

5 Conclusion: The Case for Open Source in Agricultural Biotechnology

The suggestion offered in this paper is that of an increased use of open source approaches in agricultural biotechnology, while recognizing the differences between this and other industries. By initially allowing for greater access to the workings of a biotechnological invention, greater public and scientific awareness may bolster support for new agricultural and food products. This would lead to a decreased use of dispute proceedings, often based on a lack of information or understanding. Further, it would lead to more rapid product improvement. For example, flaws in GM crops could be discovered and corrected earlier because the development process would be scrutinized, and solutions proffered, by a greater number of experts. While this has been accomplished in other selected industries, a more widespread adoption of open source agreements in biotechnology requires changes in the incentive structure inherent the proprietary, closed source approach used in that industry heretofore. While this may initially affect the process and pace of innovation, it will likely give greater credibility and legitimacy to new food products. As a result, there would be more rapid and greater acceptance, adoption and use of agricultural innovations through the reduced reliance of the global regulatory system and fewer extreme reactions such as complete moratoriums on GM consumption.

It should be kept in mind that some caution is needed with regards to the adoption of open source software agreements to biotechnology research on a wholesale level. For example, there are recent examples of individuals granting licensing to their code via an open source code model in the software context. This experience should be considered when discussing the feasibility and desirability of adopting an open source approach in the area of agricultural biotechnology. We observe further, that utilized open source code may later be patented and commercialized. Therefore, there are additional considerations of how to prevent the utilization of open sourced information to create closed sourced new innovations. It has been suggested that, even though the secondary work is distributed under a proprietary license that the accompanying initial open source code remain of that nature (Nadan 2002). In general, however, open source software models are grounded in a 30-year trial and development process. The similarities between the factors and problems behind the

[10] Although this initiative was concluded in 2007, the information published from the initiative remains widely distributed.

free software open source movement then and the agricultural biotechnology situation of today are vast. Therefore, we argue that the probability of enjoying a similar success today by applying a solution that has proven successful in the past makes open source a real and pragmatic option for policy-makers.

References

Baldwin CY (2008) Where do transactions come from? Modularity, transactions, and the boundaries of firms. Ind Corp Change 17:155–195

Bratspies R M (2003) Myths of voluntary compliance: lessons from the StarLink corn fiasco. William and Mary Environmental Law and Policy Review 27:593–599

Broothaerts W, Mitchell HJ, Weir B et al (2005) Gene transfer to plants by diverse species of bacteria. Nature 433:629–633

Feldman R (2004) The open source biotechnology movement: is it patent misuse? University of Minnesota Law School, Minn J Law Sci Technol 6:117–136

Free Software Foundation (2009) The GNU operating system: the free software definition. http://www.gnu.org/philosophy/free-sw.html. Accessed 23 Feb 2010

Goldman R, Gabriel RP (2005) Innovation happens elsewhere. Morgan Kaufmann, Burlington, MA

Graff, GD, Rausser GC, Small A (2003) Agricultural biotechnology's complementary intellectual Assets. Rev Econ Stat 85:349–63

Grossman M (2001) Biotechnology, property rights and the environment. Am J Comp Law 50:215–224

Guibault L, Hugenholtz B (2006) The future of the public domain: identifying the commons in information law. Information law series, vol 16. Kluwer Law International, The Hague

Gurbani V, Garvert A, Herbsleb JD (2006) A case study of a corporate open source development model. In: Proceedings of the 28th international conference on software engineering (ICSE), ACM Press, New York, pp 472–481

Hope J (2008) Biobazaar: the open source revolution and biotechnology. Harvard University Press, Boston, MA

International HapMap Consortium (2007) A second generation human haplotype map of over 3.1 million SNPs. Nature 449:851–862

Jefferson R (2008) Science as social enterprise: the CAMBIA BiOS initiative. In: Fitzgerald B (ed) Legal framework for e-research: realising the potential. Sydney University Press Law Books, Sidney

Lerner J, Tirole J (2005) The economics of technology sharing: open source and beyond. J Econ Perspect 19:99–120

Maurer S, Scotchmer S (2006) Open source software: the new intellectual property paradigm. NBER Working Paper 12148

Munga N, Fogwill T, Williams Q (2009) The adoption of open source software in business models: a red hat and IBM case study. Proc 2009 Ann Res Conf S Afr Inst Comput Sci Inf Technol 1:112–121

Nadan C (2002) Open source licensing: virus or virtue? Tex Intellec Prop Law J 10:349

Nagy D, Yassin A, Bhattacherjee M (2010) Organizational adoption of open source software: barriers and remedies. Commun ACM 53:148–151

O'Mahoney S (2003) Guarding the commons: how community managed software projects protect their work. Res Policy 32:1179–1198

Open Source Initiative (2010a) The open source definition. Open Source Initiative. http://www.opensource.org/docs/osd. Accessed 19 Feb 2010

Open Source Initiative (2010b) Open source licenses. Open Source Initiative. http://www.opensource.org/licenses. Accessed 19 Feb 2010

Ortí L, Carbajo RJ, Pieper J et al (2009) A kernel for the tropical disease initiative. Nat Biotecnol 27:321

Pray CE, Naseem A (2003) The economics of agricultural biotechnology research. Food and Agriculture Organization of the United Nation's Agricultural Development Economics Division (ESA) Working Paper 03-07

Raymond ES (1999) A brief history of hackerdom. In: DiBona C, Ockman S (eds) Open sources: voices from the open source revolution. OReily Media, Sebastopol, CA

Samuelson P (2006) Enriching discourse on public domains. Duke Law J 55:783–797

Winickoff D, Jasanoff S, Busch L et al (2005) Adjudicating the GM food wars: science, risk, and democracy in the world trade law. Yale J Int Law 30:81–123

Wong J (2003) Are biotech crops and conventional crops like products? An analysis under GATT. Duke Law Tech Rev 27:27–28

A Requirements Engineering Approach to Improving IT-business Alignment

Azmat Ullah and Richard Lai

Abstract Today, doing business is complex and dynamic due to rapid changes in technology and globalization of the market. Organizations require complex business structures which include: a business model, strategies, operations, etc as well as well-managed business processes (BP) which are continually improved and evaluated. However, sustaining BP is difficult due to non-alignment between information technology (IT) and other organizational components. This chapter presents a business goal-driven requirements engineering approach for IT to better understand system requirements, as requirements are generated from the business goals and provide assistance for analyzing business goals quickly.

1 Introduction

It is widely accepted that organizations are continually facing rapid changes in the business environment, particularly in relation to changes in consumer services, technologies and product lifecycles. In this context of rapid innovation and tough market competition, it is the manager's responsibility to guarantee the long term business success of the organization. Literature shows that organizations continuously update their strategies in response to market demand; for example, in the 1960s, the predominant organizational strategy was to produce quantity, in the 1970s to produce low cost products, in the 1980s to produce quality products, in the 1990s to produce products in less time, and in the twenty-first century, the strategy has changed towards offering more services [14].

Therefore, successful business organizations usually employ a structure with many complex components: first, a precise business model is used to create, deliver, and capture the organization's values; second, an efficient business strategy is required to accomplish the proposed business model; and third, the operations, which

A. Ullah (✉)
Department of Computer Science and Computer Engineering, La Trobe University, Melbourne, Australia
e-mail: a.ullah@latrobe.edu.au

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_62,

consist of further sub-components such as people, processes, information technology (IT) etc. Processes are a stable key element which helps deliver the customer's organizational requirements and reduces product cost and inefficiency. However, the method by which processes are managed is critical and business executives and managers often experience difficulties due to non-alignment between IT and other related activities. To address this difficulty, many researchers [2, 4, 6, 7, 13, 17] have proposed various solutions, but these are mostly business driven and do not provide sufficient information on the IT side. As IT has difficulty in understanding business requirements, it is unable to implement the system according to business expectations, which may lead to non-alignment in the organization.

Moreover, the development of a successful IT system not only needs an understanding of business requirements, the business activities must also be taken into account before commencing the development phase of the system, therefore goal modeling is required before implementation. A number of important modeling techniques have been explored [3, 9, 12] in regard to clarification of business goals and requirements. However, these techniques contain several weaknesses: first, they are complex for IT developers to understand; second, they provide little information on business goals; and third, they are time consuming in the analysis of business goals and unable to support a service-oriented environment.

This article presents a methodology to derive the system requirements from organizational models by means of goal modeling. Two main stakeholders participated: (1) Automatically greatened UML sequence diagrams to resolve errors or conflicts among requirements; and (2) Automatically generated UML state diagrams which represent a true picture of business goals and explain exactly what the system has to do and how it should do it.

2 Business Goals and System Requirements

Business goals can be defined as an organizational objective or target that needs to be completed. Therefore, for the successful completion of an objective, it is important to analyze the goals before implementation. This is called goal modeling and it is a useful method by which to capture and clarify the business goals of an enterprise. In other words, as one business goal can carry more than one sub-goal and every sub-goal is linked to each other, it is important that these are explored for better clarification. For example, part A of Fig. 1 shows a business that has a goal that contains a further three sub-goals: Goal A, Goal B, and Goal C and every goal is connected to each other which describes the priority of the goal.

Moreover, it is widely accepted that business goals can play a pivotal role in developing the IT system according to business expectations. The development of an effective IT system needs requirements in a way that answers the following: "Who is the stakeholder? "; "What does the system need? " and "What needs to be included in it? " Developing business goals is the only method by which to provide this kind of information, as shown in part B of Fig. 1.

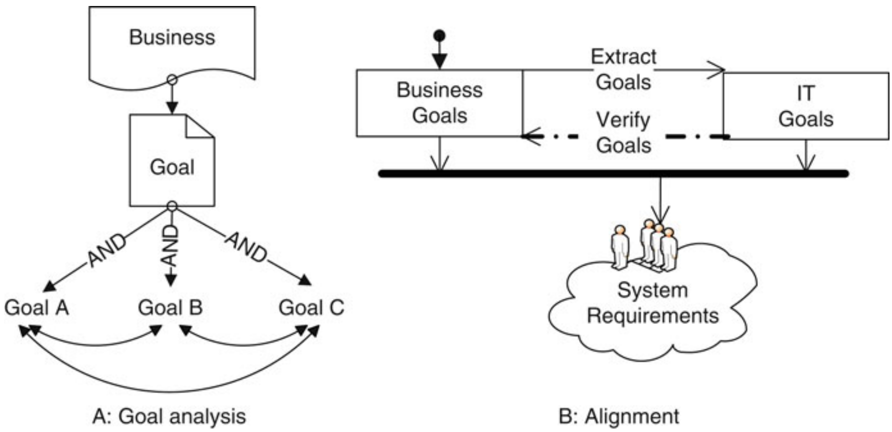


Fig. 1 Business goal analysis versus alignment

3 IT-business Alignment Concept

The term IT-Business alignment refers to the optimized synchronization between dynamic business objectives/processes and respective technological support by IT. Successful alignment has a positive influence on each component of the organization, such as business processes, structure, culture, relationships, strategies, communication etc. Moreover, in alignment, both fields of business and IT are interrelated, as IT tries to provide services at all levels of the business organization to effectively achieve goals and objectives. Achieving alignment has been a critical issue for several business firms, causing researchers and business executives to work on this issue from the early 1970s [15, 16]. However, bridging this gap has been regarded as difficult by IT researchers and business executives for several reasons: a lack of IT methodologies, differences in aims and objectives, rigid organizational structure and culture, and a communication gap between both.

Alignment has several phases, where each phase represents a specific part of the business organization, for example, internal and external phases of alignment. In the external phase, the business organization is aligned with the business partner or with other similar business organizations including clients, dealers, competitors etc. In the internal phase of alignment, business aligns with their own internal departments. This kind of alignment could be organizational phase alignment, departmental phase alignment, upper and lower phase alignment, project phase alignment, system phase alignment, etc [11].

Moreover, alignment can also be measured in terms of different aspects of the business organization such as the organization's structure, culture, strategy, and social direction. In relation to the organization's structure, researchers considered "complex structures", and "rapid changes in structure". In relation to the organization's culture, they investigated "lack of communication", "weak relationships" and "low IT belief within the organization". In relation to the organization's strategic directions, factors such as “formal business” and “formal IT strategy” were considered in terms of alignment. In relation to the social direction of the organization, factors such as “a lack of shared domain knowledge”, “a lack of IT knowledge in the business” and “a lack of business knowledge of IT” were considered.

4 The Proposed Methodology Framework

A successful business requires a set of complex management activities which include a business model, a business strategy and operations, where the business model is used to refer to how an organization creates, delivers, and captures value; the strategy is used to define and fulfill the business model; and the operations implement the overall business strategy, which consists of further sub-elements such as people, processes and technology. Processes in business operations are stable key elements which need to be well managed and fit optimally with the proposed business strategy and other related organizational activities. Every business organization needs an efficient business process to enable the business strategy to be implemented and to achieve business goals and objectives effectively [1]. However, it has long been well known that organizations continuously face problems in organizing BP.

The aim of this methodology is to resolve the issue of miss-synchronization between IT and other organizational activities. It is structured in two stages as shown in Fig. 2. Stage 1 describes the specifications of the business strategy and business infrastructure and stage 2 describes how to model the IT infrastructure, which includes several phases: Phase 1 translates the business goals using UML sequence diagrams; Phase 2 depicts the sequence diagram data into a loosely coupled format so that the information can be examined; Phase 3 analyzes the business goals in order to clarify and to resolve conflicts between goals; Phase 4 presents how to developed a graphical view of the business goals using UML state charts in regard to clarification of the system requirements for IT developers.

4.1 Business Strategy and Infrastructure

The term “modeling infrastructure” refers to the creation, delivery and capture of the organization’s values. The idea we use in this paper is based on Henderson and Venkartraman’s Strategic Alignment Model called SAM [10], a widely accepted model in the area of tackling business strategy and modeling business infrastructure. The planned methodology describes strategy by means of an organizational mission statement and by means of strategic goals. The organizational mission statement defines what the business organization is, why the organization exists, and the aims and objectives that need to be achieved, whereas strategic goals are used to represent why business processes exist and how to fulfill the organization’s mission statement.

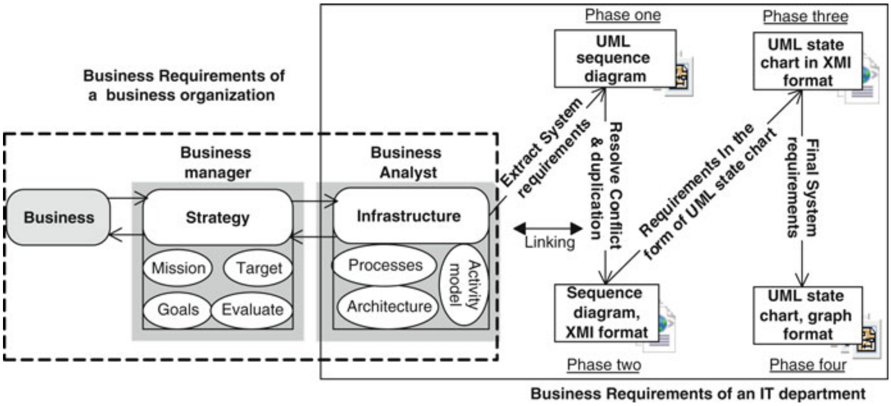


Fig. 2 The proposed methodology framework

Evaluation of strategic goals and target levels have been defined in order to measure the success of the organization, as shown in phase 1 of Fig. 2.

After the development of the business strategy and clarification of the business goals, there is a need to model the business infrastructure. The methodology divides the business infrastructure into three sub-models: (1) the Activity model represents the organizational actors and their responsibilities in the proposed organization; (2) the Resource model describes the organizational rules and regulation; and (3) the Architecture model (which consists of three different levels—the strategic level representing the organization's detailed planning process; the managerial level analyzing the proposed business strategy; and the operational level implementing the business goals).

4.2 Linking IT with Business

After the organizational infrastructure has been implemented, the methodology links business infrastructure with IT infrastructure to establish proper specifications between both these groups. Therefore, the IT system is able to reflect the organization's behavior and meet the needs of the business organization; hence IT-Business coordination can be achieved. Figure 2 shows that establishing a link between business and IT infrastructure is possible.

4.3 Modeling IT Infrastructure

The term *IT infrastructure* became popular in the mid 1990s. It refers to a set of shared IT resources that work together to achieve common goals. IT infrastructure consists of technical components, which include software, hardware, networks and telecommunication; and human components which include the technical skills of the persons, and their knowledge that is required to maintain the IT resources. In the context of BP, processes are increasingly becoming more and more complex every day and goals and objectives are changing rapidly. In a situation such as this, IT infrastructure needs to be flexible so that rapid changes in business goals and objectives can be managed [5].

We model IT infrastructure in three steps: (1) translation of business goals into common use language so that they can be easily understood by all involved; (2) analysis of business goals to resolve conflicts between business goals; and (3) derivation of system requirements to implement business goals.

4.3.1 Translation of Business Goals and Analysis

Once it has been decided which goals need to be implemented, they are translated into the UML sequence diagrams for analysis and to obtain valid system requirements to gain a true picture of the goal. A sequence diagram gives a pictorial representation of the sequence in which system objects communicate with each other to complete a business goal, where objects in the sequences are shown as rectangles and have a vertical lifeline. A lifeline is used to show when an object is created and when it is going to be destroyed. Arrows in the sequence diagram represent the messages between the objects, where each message must have a sender and receiver [8].

For an analysis of business goals, the methodology takes sequence diagrams as input in the form of XMI [18] and uses the java document object model (JDOM) to read the input XMI, line by line. Algorithm 1 of Table 1 describes how the goal is analyzed by taking a valid XMI file as an input and using a JDOM parser to extract the required information. If the JDOM becomes equal to the "*Model*" object, it extracts the model name and id, where the model represents the header of the XMI which includes the XMI version. In the header, if JDOM = "*classifier role*" object, it extracts the classifier name and id, where the classifier is the content of the XMI which is used to represent the class name of the UML sequence diagram. If JDOM = "*interaction.message*" object, it extracts the child elements of the classes, where "*interaction.message*" is the message between the two classes. Algorithm 2 of Table 2 is used to identify conflicts or errors among the classes and interactions of the sequence diagrams. If any duplication is found among classes or interactions, the program resolves it automatically.

4.3.2 Derivation of System Requirements

For successful IT system implementation, it is important that goals should be defined in a way that clarifies the system requirements. Therefore, we convert business requirements from UML sequence diagrams to a UML state chart automatically. State charts depict exactly what the business requirements are and they also explain

Table 1 Automatic goals analysis and generation of state chart

Algorithm 1	Algorithm 3
1. **Start**	1. **Start**
2.Define:	2. Define:
3. JDOM: Parser to read XMI;	3. JDOM: Parser to read XMI;
4. Model: Used to store version of the XMI	4.interaction.message: Interactions in input file;
5. ClassifierRole: Represents classes in XMI;	5. Let Fout be the file, where output store;
6. Interaction.message: Represent interaction	6. **if** Fout does not exit **then**
between the classes of the XMI;	7. create new file;
7. Let Fs ={Fs1, Fs2, … Fsn} be the valid XMI.	8. **end if**;
8. **For each** line Li of file Fs Li ∈ Fs	9. Let Let Fs = {Fs1, Fs2, … Fsn} be the valid XMI file.
9. **If** (JDOM = = Model) **then**	10. **for each** line Li in input file Li ∈ Fin
10. get model name, id;	
11. **else If** (JDOM = = ClassifierRole) **then**	11. **if** (JDOM = = interaction.message, i) **then**
12. get Classifier name, id;	12. temp.value <= i.value;
13. call algorithm 2;	13. temp.name <= name
14. **else If** (JDOM = = interaction.message) **then**	14. temp.name <= i.name;
	15. get sender value SV and receiver value RV;
15. get interaction name, id;	16. **end if**;
16. call algorithm 2;	17. **if** (i.value= = SV) **then**
17. **end if**; **end if**; **end if**;	18. write as state of the state chart in Fout;
20. **for each** interaction I of file Fs I ∈ Fs	19. **else if** (i.value= = RV) **then**
21. temp.id <= I.id;	20. write as label of the state chart in Fout;
22. **if** (I = = classifier.id) **then**	21. **end if**;
23. get classifiers name (sender and receiver);	22. **end if**;
24. **end if**; **end for**;	23. **end for**;

Table 2 How to resolve errors or conflicts

Algorithm 2
1. **For each** classifier Fs and interaction.message of Fs **do**
2. **If** (New_Classifier CN.name = = Pren_Classifie Cp.name) **then**
3. Report conflicts arise "class duplication";
4. **else If** (New_Classifier CN.name = =empty) **then**
5. Report conflicts arise "Null class name";
6. **end if**; **end if**;
8. **If** (New_interaction Ni .name= = Prev_interaction Ip.name) **then**
9. Report conflicts arise "interaction duplication";
10. **else If** (New_interaction Ni .name= = empty) **then**
11. Report conflicts arise "Null interaction name" ;
12. **end if**; **end if**; **break**;

what the system has to do. Algorithm 3 of Table 1 describes how the methodology works.

The generated state chart at this stage gives a true picture of the system requirements. After generation of all state charts from the proposed business goal, similar goals that represent the specific part of the system save it into one package and send it to the developer for implementation. The developer first checks the package for ambiguity; if ambiguity is detected, the developer sends it back to the system analyst for further modification. However, if the developer does not find any errors, the package is forwarded to the implementation team.

5 Conclusion and Future Work

Information technology has become an essential part of business organizations and this essentiality is increasing every day. However, sustaining IT-Business alignment is always hard, due to unclear business requirements among IT people. This article presents a methodology to derive the system requirements from organizational models by means of business goal modeling. The concept of modeling the business infrastructure was derived from SAM [10], and other stakeholders also participated such as UML sequence diagrams, XMI format, and UML state charts.

The methodology was successfully implemented and bringing the following benefits to both business and IT infrastructures: first, it helps IT developers to understand the business goals as the goals have been analyzed and clarified according to the system demands before sending it to the IT developers for implementation; second, it allows IT developers to implement the IT system according to the organization's needs which improves the IT belief among the business executives and managers; third, it ensures the business organization's success in service-oriented environments as the business goals are automatically analyzed, which alternately influences the process of IT-Business alignment and has a positively influence on the management of business processes, since the system requirements are inherited from organizational goals. Possible future investigation will involve validating the methodology with business industry process in order to enhance the sustainability of the methodology.

References

1. Armistead C, Harrison A, Rowlands P (1995) Business process re-engineering: lessons from operation management. Int Oper Prod Manag 15:46–58
2. Broadbent M, Weill P (1993) Improving business and information strategy alignment: learning from the banking industry. IBM Syst J 32(1):162–179
3. Bubenko J, Persson A, Stirna J (2001) EKD user guide. http://www.dsv.su.se/~js/ekd_user_guide.html. Accessed 11 Jan 2010
4. Burn JM (1993) Information systems strategies and the management of organizational change—a strategic alignment model. J Infor Technol 8(4):205–216

5. Byrd TA, Turner DE (2000) Measuring the flexibility of information technology infrastructure: exploratory analysis of a construct. J Manag Infor Syst 17:167–208
6. Chan YE (2002) Why haven't we mastered alignment? The importance of the informal organization structure. MIS Q Execut 2(1):97–112
7. Feeny DF, Edwards BR, Simpson KM (1992) Understanding the CEO/CIO relationship. MIS Q 16(4):435–448
8. Fowler M, Scott K (1999) UML distilled: a brief guide to the standard object modeling language. Pearson Education Inc, Boston, MA
9. Gordijn J, Yu E, Raadt BVD (2006) E-service design using i* and e3 value modeling. IEEE Softw J 23(3):26–33
10. Henderson C, Venkatraman N (1992) Strategic alignment—a model for organizational transformation through information technology. Massachusetts Institute of Technology, Cambridge, MA
11. Henderson JC, Venkatraman N (1993) Strategic alignment-leveraging information tech nology for transforming organizations. IBM Syst J 32(1):4–16
12. Jacobs S, Holten R (1995) Goal-driven business modelling—supporting decision making within information systems development. In: Proceedings of the conference on organizational computing systems, Milpitas, CA pp 96–105
13. Kearns GS, Lederer AL (2000) The effect of strategic alignment on the use of IS-based resources for competitive advantage. J Strategic Inf Syst J 9(4):265–293
14. Lindsay A, Downs D, Lunn K (2003) Business processes—attempts to find a definition. Infor Softw Technol J 45:1015–1019
15. Luftman JN, Lewis PR, Oldach SH (1993) Transforming the enterprise: the alignment of business and information technology strategies. IBM Syst J 32(1):198–221
16. McKeen D, Smith A (2003) Making IT happen: critical issues in IT management. Wiley, London
17. Reich BH, Benbasat I (2000) Factors that influence the social dimension of alignment between business and information technology objectives. MIS Q 24(1):81–113
18. Stevens P (2003) Small-scale XMI programming: a revolution in UML tool use. Autom Softw Eng 10:7–21

Adapting the Lesk Algorithm for Calculating Term Similarity in the Context of Requirements Engineering

Jürgen Vöhringer and Günther Fliedl

Abstract Calculating similarity measures between schema concept terms has turned out to be an essential task in our requirements engineering research. The reason is that automatic integration of source texts and terms in the requirements engineering domain presupposes decisions about the similarity and the conflict potential of linguistic units, i.e. by default words. After shortly discussing several WordNet based similarity measures, we propose the Lesk algorithm as the most suitable choice for the integration task. Our test results show that both the Lesk algorithm and the underlying lexicon (i.e. WordNet) can be optimized for engineering purposes. We describe in detail how Lesk and WordNet can be applied for term conflict recognition and resolution during the engineering workflow.

1 Introduction

For supporting systematic and requirements oriented ontology engineering we previously researched and proposed linguistic guidelines for structuring concept and property notions in Web Ontology Language (OWL) represented ontologies (Fliedl et al. 2008). But these guidelines only support ontology generation if domain expertise is sufficiently available in a clearly decoded manner (Vöhringer et al. 2010). Moreover our previous approach presupposed working with a single, clearly defined domain ontology. In requirements engineering however, frequently several different ontologies are involved, because information is collected from various sources. Thus for constructing a single comprehensive ontology, certain non-trivial integration steps must be performed (Bellström and Vöhringer 2009).

J. Vöhringer · G. Fliedl (✉)
Institute of Applied Informatics, Alpen-Adria-Universität Klagenfurt, Klagenfurt, Austria
e-mail: juergen@ifit.uni-klu.ac.at

G. Fliedl
e-mail: fliedl@ifit.uni-klu.ac.at

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_63, © Springer Science+Business Media, LLC 2011

This paper focuses on an automatic solution strategy for ontology integration problems in the context of requirements engineering. This type of integration presupposes concept term comparison, conflict recognition and conflict resolution. Usually the integration steps are performed manually. For our computer-aided approach (Bellström and Vöhringer 2009) the inclusion of term-related similarity measures is indispensable: we apply them to identify distances between relevant concept terms and therefore find hints for potential conflicts.

The rest of the paper is structured as follows: Sect. 2 contains a general discussion of potentially relevant WordNet (Miller 1995) related similarity measures. The compared algorithms are all taken from the CPAN WordNet::Similarity package (Pederson et al. 2004). In Sect. 3 we describe the Lesk algorithm in more detail and discuss its gloss overlapping strategy. Then we present our optimized Java Lesk implementations including the comparison of some exemplary result scores. In Sect. 4 we show how we apply Lesk and WordNet in the requirements engineering domain. The paper closes with a conclusion and an outlook on open research issues.

2 WordNet Based Similarity Measures

The lexical database WordNet offers a particularly extensive taxonomy for the English language. It is freely available and therefore widely used in similar research projects. Its big repository of natural language gloss texts makes WordNet a suitable database for all kinds of natural language optimization tasks. The WordNet::Similarity package (Pederson et al. 2004) offers possibilities for measuring concept similarity based on WordNet's internal structure and/or word sense definitions, among them the following:

- Wu & Palmer (wup)
- Resnik (res)
- Hirst & St Onge (hso)
- Leacock & Chodorow (lch)
- Lesk (lesk)

The *Wu & Palmer metric* was first suggested in (Wu and Palmer 1994). It presupposes a lexicon or ontology with a taxonomic structure. In a first step the least common subsumer (LCS) is determined, i.e. the first common parent of the compared concepts in the taxonomy. A longer distance between the concepts and their LCS means a lower similarity.

For the *Resnik metric* (Resnik 1995) the information content (IC) of the LCS probability is determined, where the LCS probability is defined as the chance of encountering the LCS-concept in a corpus. The IC is defined as the negative logarithm of the concept's probability. This means the IC declines for concepts that are more probable to occur in the corpus. Since they subsume all other concepts, top level concepts naturally have a probability of 1 and an IC of 0.

The *Hirst & St Onge metric* (Hirst and St-Onge 1998) allows measuring the similarity of two concepts by determining the length of the interjacent taxonomy path. Three different kinds of concept-connecting paths can be distinguished: extra-strong, strong and medium. Extra-strong paths exist between two equivalent concepts. Strong path are identified by a direct connection between two different concepts. Medium-strong paths exist between two indirectly connected concepts. In the latter case the number of direction changes is relevant for determining the concept similarity. Direction changes occur each time when switches between upwards-paths (generalizations), downward-paths (specializations) and horizontal paths (other relationships between concepts) take place in a medium-strong connection.

Leacock and Chodrow (Leacock and Chodorow 1998) also propose path length counting for determining concept similarity. Here the path length is computed by counting the nodes between the compared concepts. The Leacock and Chodrow metric only considers nodes from hypernymy-relationships.

The *Lesk* metric (Banerjee and Pederson 2002) is a context based approach. This metric presupposes a lexicon which differentiates word senses and provides detailed definitions for each meaning. Unlike other WordNet based similarity measures neither LCS nor path length are required. For calculating the Lesk similarity, the definitions for both involved concepts are provided. Then a numerical estimation of the degree of intersection between the concept definitions is calculated by counting the word overlap. See Sect. 3 for a more detailed description of specific Lesk implementations.

The above mentioned similarity measures are compared in several publications. In (Warin et al. 2004) an experiment was performed comparing five different similarity measures (e.g. Jiang-Conrath, Lin and Wu&Palmer). Leacock and Chodrow was singled out as the best performing score. Other studies have arrived at different conclusions however, e.g. (Budanitsky and Hirst 2001) evaluated the same measures and concluded that the Jiang-Conrath measure performed best overall. Taking the diverging results into account we assume that none of the compared measures is universally preferable. Due to their inner mechanics most metrics strongly depend on the taxonomy structure of the lexicon (i.e. height, coverage, sub-tree structure), and their usefulness varies with respect to the specific taxonomy and concept structure.

For our purposes we prefer using the Lesk algorithm. In contrast to other algorithms, Lesk does not limit the similarity calculation on hypernymy-relationships (see Table 1); it rather includes all kinds of term relationships, including hyponymy. Although the Hirst and St Onge algorithm also utilizes all term relationships, we prefer Lesk anyhow because it focuses on gloss comparison (see Table 2). This fact makes the Lesk algorithm easier to adapt and more robust against deficits in the taxonomic structure of WordNet.

Table 1 Similarity algorithms according to Pederson et al. (2004)

	wup	res	hso	lch	lesk
IC of LCS		X			
Path-length	X			X	
Not exclusively... hypernymy related			X		X

Table 2 Similarity algorithms based on required lexicon information elements

	wup	res	hso	lch	lesk
LCS	X	X			
Path/distance			X	X	
Gloss					X

Table 3 Typical WordNet glosses for term relationships

Bus#n#1		
Gloss type	Description	Gloss value
Example	Example of usage	"He always rode the bus to work"
Gloss	Word sense definition	A vehicle carrying many passengers; used for public transport
Hype gloss	Bus is a kind of...	*Public transport:* conveyance for passengers or mail or freight
Hypo gloss	... Is a kind of bus	• *Minibus:* a light bus (4-10 passengers) • *School bus:* a bus used to transport children to or from school • *Trolleybus:* a passenger bus with an electric motor that draws power from overhead wires
Holo gloss	Bus is a part of...	*Fleet:* group of motor vehicles operating together under the same ownership
Mero gloss	... Is a part of bus	• *Roof:* protective covering on top of a motor vehicle • *Window:* a transparent opening in a vehicle that allow vision out of the sides or back; usually is capable of being opened

3 Optimizing the Lesk Algorithm

Our own Java based Lesk implementation is based on the CPAN WordNet::Similarity implementation of the Lesk algorithm. We use it for conflict recognition and resolution in multi source integration for requirements oriented ontology engineering. In this algorithm the following glosses are identified for each compared word: word gloss (definition), examples, hypernyms, hyponoyms, meronyms, holonyms. Permutations of the various relationship glosses are arranged and tested for possible overlaps.

Table 3 lists examples of word relationship glosses extracted from WordNet. Note that the words written in italics are not part of the actual gloss. They are added here for clarity reasons, because they denote the corresponding concepts of the gloss.

The score for each gloss-pair is calculated as the sum of the squared word count of each overlap. Squaring the word count is a strategy for prioritizing longer matches.

Table 4 Example of standard Lesk overlaps

Tracing Lesk comparison car#n#1—bus#n#1			
Car#n#1	Bus#n#1	Overlap	Score
Example	Gloss	1 × "a"	1
Gloss	Gloss	1 × "a" 1 × "vehicle"	2
Gloss	Mero gloss	1 × "usually" 1 × "a motor vehicle"	10
Hypo gloss	Gloss	1 × "passengers" 1 × "a vehicle" 1 × "for" 1 × "used"	7
Mero gloss	Mero gloss	1 × "a transparent opening in a vehicle that allow vision out of the sides or back usually is capable of being opened" 1 × "protective covering on top of a motor vehicle"	505
Total score			615

The score of all compared gloss-permutations is accumulated for the total similarity score. Table 4 shows exemplary similarity scores generated by applying the described standard Lesk algorithm. Comparing the example gloss of car#n#1 and the (descriptive) gloss of bus#n#1 yields a single overlap of the length 1. The score for this overlap is 1^2, therefore 1. The hyponym gloss of car#n#1 and the (descriptive) gloss of bus#n#1 have four overlaps. Three of them have the length 1 and one of the overlaps ("a vehicle") consists of two words, i.e. it has the length 2. The score is calculated as follows: $3*1^2+1*2^2$, therefore 7. The total similarity score for car#n#1 and bus#n#1 is 615.

The standard Lesk algorithm can be optimized with regard to internal and external factors:

- Internal factors (Optimization of the algorithm):
  - Partial filtering of stop words
  - Word reduction via stemming
  - Normalization based on gloss length
- External factors (Optimization of the lexicon, e.g. WordNet):
  - Improvement of glossary quality and quantity via completion and substitution of certain keywords
  - Restructuring of taxonomy
  - Guidelines for gloss extension in specific domains

In our Java implementation we use stop word filtering for deleting word grams containing a certain empirically motivated percentage of non-content words. To this end we use a stop word list generated from WordNet. Our strategy includes filtering all gloss overlaps containing at least a predefined percentage of stop words. Single stop word overlaps are assigned a 100% stop word quotient and they are therefore obligatorily filtered out. A similar stop word filtering approach was discussed in (Banerjee and Pederson 2002). For stemming purposes we use an implementation of the extended Porter stemmer (Willett 2006). We argue that stemming is necessary for identifying matches of inflected word variants.

Beyond that the Lesk score could also be normalized by the size of the glosses. We ignore this normalization because the differences between non-normalized scores are more expressive with respect to the intended extensions and optimizations of domain specific WordNet glosses. The expressiveness of the normalized scores is diminished because the importance of matches in small glosses is boosted in an unnatural way.

In the following the optimization effect is shown by comparing the word pairs car#n#1—motorcycle#n#1 and car#n#1—bicycle#n#1. Although those words may not be representative for our domain, we chose them because their relative similarity distance is intuitively comprehendible. The original Lesk score lists "car"-"bicycle" as more similar than "car"-"motorcycle" (a score of 300 vs. 237), while our optimized Lesk implementation yields the scores 115 for the word pair "car"-"bicycle" and 181 for the word pair "car"-"motorcycle". Thus our algorithm better reflects the obvious higher similarity of "car" and "motorcycle".

The difference in the results can be easily explained by highlighting some details of our internal optimization. For example, comparing the hyponym-meronym glosses of the car#n#1 and bicycle#n#1 results in a standard Lesk overlap score of 44. The score is calculated on the following overlaps: 1× "that is", 1× "with a", 1× "on a", 10× "a", 4× "the", 3× "with", 3× "to", 2× "that", 2× "or", 1× "from", 1× "seat", 1× "by", 1× "for", 1× "usually", 1× "one", 1× "used", 1× "of". In our approach we automatically filter out all the matches consisting of 100% stop words. This yields 1× "seat" as the only remaining overlap and thus an optimized score of 1.

By comparing the meronym-glosses for the same word pair we receive a Lesk standard score of 28 based on the following overlaps: 1 × "water or mud", 1 × "of a", 1 × "the wheels", 1 × "above the", 1 × "a", 1 × "the", 1 × "or", 1 × "from", 1 × "thrown", 1 × "to", 1 × "by". The optimized algorithm returns nearly the same result (score 27), but is calculated differently: the matches 2 × "the", 1 × "a", 1 × "or", 1 × "from", 1 × "to", 1 × "abov" are all ignored, since they consist of stop words. The matches 1 × "water or mud", 1 × "thrown", 1 × "protect" are accepted since they all contain content-words. The match 1 × "the wheel of a" is new and made possible because the content words of the glosses were stemmed prior to the comparison. Without stemming the match wouldn't have been found, since the "bicycle"-meronym contains the string "the wheel of a bicycle" and the "car"-meronyms contain the string "the wheels of a vehicle".

Supported by the results of our experiments we argue that the results of the optimized Lesk algorithm are much more meaningful than the standard Lesk score.

The optimization with respect to external factors (e.g. filling gaps in WordNet glosses) is a long and difficult process that goes beyond the scope of this paper. However, for demonstration purposes we completed some exemplary glosses of two entries ("motorcycle", "bicycle") used in our similarity experiments. For extending the example glosses we adapted some suitable meronym gloss texts from related words ("car"). The extensions of the standard WordNet glosses are listed in Table 5.

Table 5 Filling gaps in WordNet Glosses

Concept	Extension of WordNet Standard Glosses for meronyms
Car	WordNet Standard Glosses (not extended)
Bus	WordNet Standard Glosses (not extended)
Motorcycle	WordNet Standard Glosses *extended by*
	• *Motorcycle engine*: the engine that propels a motorcycle
	• *Motorcycle horn:* a device on a motorcycle for making a warning noise
	• *Motorcycle mirror*: a mirror that the driver of a motorcycle
	• *Gasoline engine:* an internal-combustion engine that burns gasoline
Bicycle	WordNet Standard Glosses *extended by*
	• *Bicycle horn:* a device on a bicycle for making a warning noise
	• *Bicycle mirror:* a mirror that the driver of a bicycle can use

Table 6 Standard Lesk

Pair	Score	Rank
Car-bicycle	300	2
Car-motorcycle	237	3
Car-bus	739	1

Table 7 Internally optimized Lesk

Pair	Score	Rank
Car-bicycle	115	3
Car-motorcycle	181	2
Car-bus	688	1

Table 8 Internally and externally optimized Lesk

Pair	Score	Rank
Car-bicycle	198	3
Car-motorcycle	321	2
Car-bus	688	1

Summarized results of our internal and external Lesk optimizations are shown in Tables 6, 7 and 8. It should be obvious that the results in Table 6 don't adequately reflect the intuitive similarity-ranking concerning form and functionality of the involved entities. Table 7 shows that the ranking is already correctly established through our internal optimization procedure. However the score distances in this table are not yet entirely intuitive. Table 8 shows the adjusted scoring results after filling obvious lexicon gaps.

Key word substitution, taxonomy restructuring and defining guidelines for gloss extension are other tasks for optimizing external aspects for the Lesk algorithm. Discussing these tasks is out of scope of this paper. Nevertheless our first optimization results give reason to hope that these steps could lead to further improvements of the Lesk algorithm in a broader sense.

4 Using the Lesk Algorithm for Conflict Resolution During Requirements Engineering

In Table 9 the nouns from two different requirements sentences ("*For hemodialysis patients, the circulation must be monitored at a specific time and date"* and *"Blood pressure measurements are performed before each dialysis session.*") were filtered out. For this purpose we used standard linguistic filtering techniques like tagging and chunking (Vöhringer et al. 2010). We identified potential conflicts by comparing the nouns of both sentences using our optimized Lesk algorithm. All involved nouns are taken as candidates for relevant concept terms in a certain domain ontology. We propose that the word pairs with a Lesk score lower than 500 are good concept candidates without any conflict potential. The threshold of 500 was derived as best practice from our tests. Word pairs with a score of higher than or equal to 500 are classified as potential conflict pairs. Thus they are input for further processing to resolve the conflict on the next level. On the higher level our engineering workflow provides that those terms from each similarity pair that are deeper in the WordNet hypernymy hierarchy should be selected as ontology concept names.

As can be seen in Table 9, only two words from the test sentences show a similarity score of at least 500 and therefore need conflict resolution. "Hemodialysis" is obviously a specialization of "dialysis" and thus deeper in the hypernymy hierarchy. The relationship between "circulation" and "blood pressure" is not as intuitive, but looking into the WordNet taxonomy yields a clear answer:

Hypernymy hierarchy for "blood_pressure":

blood_pressure (level 7) $\geq$ *pressure (level 6)* $\geq$ *physical (level 5)* $\geq$ *natural phenomenon (level 4)* $\geq$ *phenomenon (level 3)* $\geq$ *process (level 2)* $\geq$ *physical entity (level 1)* $\geq$ *entity (level 0)*

Hypernymy hierarchy for "circulation":

circulation (level 6) $\geq$ *organic phenomenon (level 5)* $\geq$ *natural phenomenon (level 4)* $\geq$ *phenomenon (level 3)* $\geq$ *process (level 2)* $\geq$ *physical entity (level 1)* $\geq$ *entity (level 0)*

Table 9 Using Lesk scores for identifying potential conflict terms in compared example sentences

Word 1	Word 2	Optimized score
Hemodialysis	Blood_pressure	2
Hemodialysis	Measurement	4
Hemodialysis	*Dialysis*	*648*
Hemodialysis	Session	0
Patient	Blood_pressure	11
Patient	Measurement	4
Patient	Dialysis	5
Patient	Session	1
Circulation	*Blood_pressure*	*1,300*
Circulation	Measurement	14
Circulation	Dialysis	15
Circulation	Session	0

Because "blood pressure" is a more specialized concept term than "circulation", its distance to the top level concept ("entity") term is higher. For that reason "blood pressure" is chosen as the concept term in our ontology.

To sum up, we propose a twofold strategy for finding new concept terms in domain ontologies. On the deeper level we identify words pairs regarding the similarity score decoding their conflict potential. On the higher level we prioritize regarding concept positions in the WordNet hypernymy hierarchy for identifying the ultimate concept term.

5 Conclusion and Future Work

To sum up in this paper we argued for the Lesk algorithm as the most useful WordNet based similarity measure for the integration task in the requirements engineering domain. We discussed some shortcomings of current Lesk implementations and argued for optimizations of both internal and external aspects. Thus we described our Java Lesk implementation and tested it on exemplary word pairs with intuitively comprehendible relative similarity distances. We additionally outlined how the Lesk similarity measure can be used for automatized conflict recognition and resolution in the context of ontology integration in the requirements engineering phase. For future work an extensive experimental study is planned for the evaluation of our intuitive and theoretically motivated hypotheses. Our preliminary results show that further research on this topic is expedient, both for the automated support of ontology based requirements engineering and for domain-specific guided improvement of WordNet.

References

Bellström P, Vöhringer J (2009) Towards the automation of modeling language independent schema integration. In: International conference on information, process, and knowledge management, eKnow 2009, Cancun, Mexico, pp 110–115

Budanitsky A, Hirst G (2001) Semantic distance in WordNet: an experimental, application-oriented evaluation of five measures. In: Workshop on WordNet and other lexical resources, 2nd meeting of the North American chapter of the Association for Computational Linguistics, Pittsburgh

Banerjee S., Pederson T (2002) An adapted Lesk algorithm for word sense disambiguation using WordNet. In: Proceedings of the 3rd international conference on intelligent text processing and computational linguistics, Tokyo, Japan, pp 136–145

Fliedl G, Kop C, Vöhringer J (2008) Guideline based evaluation and verbalization of OWL class and property labels. Data Knowl Eng 69(4):331–342 (April 2010)

Hirst G, St-Onge D (1998) Lexical chains as representations of context for the detection and correction of malapropisms. In: WordNet: an electronic lexical database (Language, Speech, and Communication). MIT Press, Cambridge

Leacock C, Chodorow M (1998) Combining local context and WordNet similarity for word sense identification. In: Fellbaum. MIT Press, Cambridge, pp 265–283

Miller GA (1995) WordNet: a lexical database for English. Commun ACM, 38:39-41

Pederson T, Patwardhan S, Michelizzi J (2004) Wordnet::similarity—measuring the relatedness of concepts. In: Proceedings of the 19th national conference on artificial intelligence, San Jose, pp 1024–1025

Resnik P (1995) Using information content to evaluate semantic similarity in a taxonomy. In: Proceedings of international joint conference for artificial intelligence (IJCAI-95), Montreal, pp 448–453

Warin M, Oxhammar H, Volk M (2004) Enriching an ontology with WordNet based on similarity measures. In: MEANING-2005 Workshop, Trento

Willett P (2006) The Porter stemming algorithm: then and now. Electron Libr Inform Syst 40(3): 219–223, ISSN 0033-0337

Wu Z, Palmer M (1994) Verb semantics and lexical selection. In: Proceedings of the 32nd annual meeting of the association for computational linguistics, Las Cruces

Vöhringer J, Gälle D, Fliedl G, Kop C, Bazhenov M (2010) Using linguistic knowledge for fine-tuning ontologies in the context of requirements engineering. International Journal of Computational Linguistics and Applications, Vol. 1, No. 1-2 2010, in: chapter 5 Applications, ISSN 0976-0962, Bahri Publications

SysML Parametric Diagrams in Business Applications

Stanislaw Wrycza and Bartosz Marcinkowski

Abstract The Unified Modeling Language (UML) has been extensively used in the systems engineering domain. It has become a commonly accepted standard. However, not all expectations of system engineers have been met by UML. Therefore, a new language, based on UML but strongly supplemented by notions specific for the systems engineering field, has been proposed by the International Council on System Engineering (INCOSE) and Object Management Group (OMG). New types of diagrams specific for the technical spheres of systems engineering have been proposed and included in a standard set of 9 types of SysML diagrams. While they are successfully used in systems engineering, systems analysts have gone the opposite way—they have experimented with the appropriateness of SysML diagrams for applications in business and administration organizations. In this chapter, a case study of inventory control systems in industrial and commercial organizations, originating from business principles, theory of prognosis and operations research, is used to verify the usefulness of one of the leading and most characteristic SysML diagrams, i.e. the Parametric Diagram. Following the Introduction, the role of SysML in the development of inventory control systems is outlined. In Sect. 3, a Block Definition Diagram as a basis for parametric analysis is specified. Section 4 provides selected case study results. The chapter is finalized with Conclusions.

1 Introduction

The popularity of the Unified Modeling Language (Booch et al. 2004; Dennis et al. 2005; Eriksson et al. 2004; Wrycza et al. 2005), which has become a kind of a standard in Object-Oriented Information Systems Modeling as well as Software Engineering (OMG UML 2009), has inspired system engineers to elaborate and implement a non-software-centric, domain language—i.e. the Systems Modeling

B. Marcinkowski (✉)
Department of Business Informatics, University of Gdansk, Sopot, Poland
e-mail: bartosz.marcinkowski@gmail.com

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_64, © Springer Science+Business Media, LLC 2011

Language (OMG SysML 2008). SysML has become the subject of several publications, such as Weilkiens (2007), Friedenthal et al. (2008) or Holt and Perry (2008). SysML was designed to satisfy and support specific requirements of systems engineering. It was an adaptation and extensive modification (comp. Bock 2006) of the official version of UML (2.x) into the semantics of a new, graphical, diagrammatic language having rich possibilities of expressing systems engineering requirements, which was the natural way of conduct (more INCOSE 2007). Such profiling mechanism is one of the fundamental features of Unified Modeling Language, resulting in development of numerous domain-specific extensions (eg. Przybyłek 2008), profiles or complete languages. SysML may be used to support problem solving for industries like space industry, automation, military, telecommunication, biotechnology, nanotechnology, power industry, chemistry, other technical industries, healthcare or business.

Incorporating SysML into planned projects allows system engineers to create a natural link, combining different (often distant) disciplines into a holistic entity, regardless of hardware, personnel and other determinants. Language development has been coordinated by three well-known organizations:

- OMG (Object Management Group),
- INCOSE (International Council on System Engineering),
- ISO (International Organization for Standardization).

As a result of the workgroups' efforts, the Systems Modeling Language was developed and established. Formally, its specification was released in 2006 and then upgraded to version 1.1 in 2008. Although SysML concentrates on technical applications, it also attracts IT professionals involved in business applications and information systems modeling (Hause 2009; Wrycza and Marcinkowski 2009a). This research track is further continued in this chapter.

Following the Introduction, the role of SysML in the development of inventory control systems is outlined. In Sect. 3, a Block Definition Diagram as a basis for parametric analysis is specified. Section 4 provides selected case study results, being the application of Parametric Diagrams in regard of inventory control. This chapter is finalized with conclusions.

2 SysML in the Development of Inventory Control Systems

Inventory control systems have always been in the centre of Management Information Systems exploited in business and administration organizations. They are one of the most crucial modules of ERP (Enterprise Resource Planning) systems. Such systems require continuous improvement in respect of progress in management (JIT, Just in Time), forecasting and operations research upon which every inventory system is based. SysML provides conditions and opportunities for development and strict formal specification of requirements, structure and dynamics of inventory control systems. The mathematical formulas composed may be expressed explicitly in accordance with SysML constraints.

SysML supports information systems modeling by using the following nine diagrams:

- Requirement Diagram;
- Use Case Diagram;
- Activity Diagram;
- Sequence Diagram;
- State Machine Diagram;
- Block Definition Diagram;
- Internal Block Diagram;
- Parametric Diagram;
- Package Diagram.

Most of the above listed diagrams are adequate for modeling inventory control systems. However, specific formal mathematical equations and formulas may be verbalized in accordance with the constructs and principles of Parametric Diagrams. SysML Parametric Diagrams enable clear and precise expression of mathematical formulas, used in forecasting and ordering in inventory control systems. Basic modeling categories used in modeling system parameters include: constraint blocks, constraint properties, constraint parameters, binding connectors as well as measures of effectiveness. Parametric Diagrams are often used to perform a trade-off analysis, which enables a system modeler to specify and compare various variants of a target software or hardware solution.

3 Block Definition Diagrams as a Basis for Parametric Analysis

An inventory control system comprises the following main modules and procedures:

- Materials/commodities evidence;
- Material sales forecasting/prognoses;
- Materials/commodities ordering;
- Accountancy.

The second and third of the above listed modules are of special research interest in this chapter. Therefore, the following methods are included in the considerations:

- Exponential smoothing for forecasting;
- Optimal purchase order calculations for ordering module.

The results of sales forecasting for each material/commodity item are the input for the calculation of optimal purchase orders for those items.

A Parametric Diagram does not initiate analysis and design discipline (IBM 2009), instead it acts as one of the diagrams that finalize it, bridging specification and

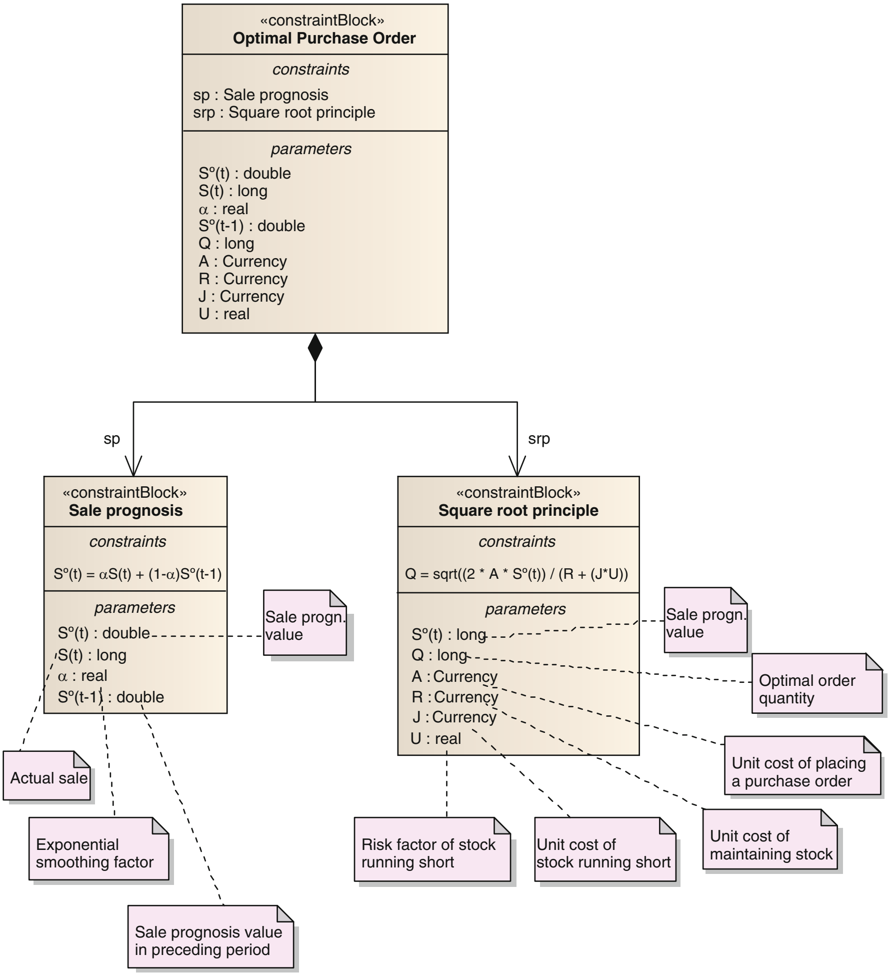


Fig. 1 Specification of a constraint block with a Block Definition Diagram

software implementation. The diagrams can be applied to specific software after creating at least Block Definition Diagrams. With respect to inventory control systems, a Block Definition Diagram, presenting the interrelationships and dependencies between respective components of inventory control systems, is shown in Fig. 1.

As a result of inventory control procedures, optimal order quantity should be calculated and an order placed by the system. The *Optimal Purchase Order* constraint block is composed of two subsequent constraint blocks: *Sale prognosis* and *Square root principle*. The latter two constraint blocks have mathematical constraints assigned as well as all the necessary parameters for calculating the formulas corresponding to the constraints enumerated in the parameters section. Mathematical constraints of the respective constraint blocks were defined as follows:

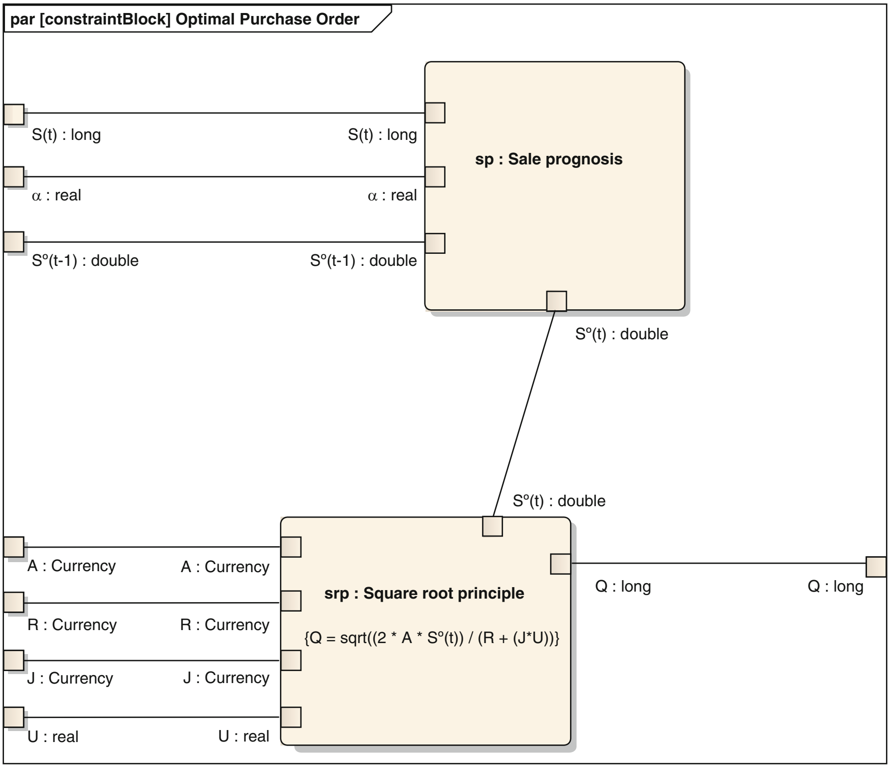


Fig. 2 Parametric Diagram for optimal purchase order

- Sale prognosis—$S^\circ(t) = \alpha S(t) + (1 - \alpha)S^\circ(t - 1)$;
- Square root principle—$Q = \sqrt{(2 * A * S^\circ(t)) \,/\, (R + (J * U))}$.

Naturally, this type of calculations are accomplished first of all in respect of products of substantial significance to a company, taking into account revenue and profit. Therefore, it is a common practice to provide segmentation for such calculations. Only the products at the top of such segmentation are the subject of advanced forecasting and interconnected ordering formulas.

4 Parametric Diagrams—Optimal Purchase Order Case Study

On the basis of a Block Definition Diagram, Parametric Diagram(s) may be elaborated (Wrycza and Marcinkowski 2009b). Figure 2 presents a Parametric Diagram based on the Block Definition Diagram illustrated in Fig. 1.

The diagram shows the components of the *Optimal Purchase Order* constraint block, i.e.:

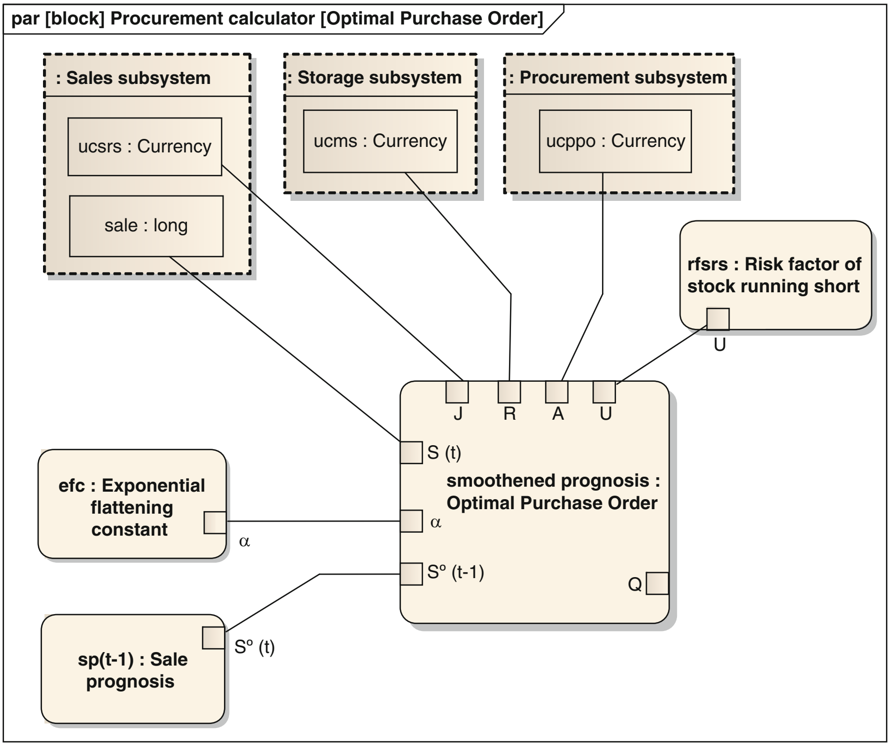


Fig. 3 Binding values for individual parameters

- *sp* (constraint property being an application of the *Sale prognosis* constraint block in the specific context);
- *srp* (application of the *Square* root *principle* constraint block).

The result of sale prognosis calculation ($S°(t)$) acts as one of the inputs of the square root principle. In order to calculate the sale prognosis value, the following constraint parameters are used:

- $S(t)$—actual sale;
- α—exponential smoothing factor;
- $S°(t\text{-}1)$—sale prognosis value in preceding period.

Apart from sale prognosis calculation, the calculation of the optimal order quantity (Q) is feasible by using other input data:

- U—risk factor of stock running short;
- J—unit cost of stock running short;
- R—unit cost of maintaining stock;
- A—unit cost of placing purchase order.

In this way, specification of optimal order quantity is possible. The values discussed above may have its source in the current subsystem/module, other subsystems or other constraint properties (see Fig. 3). Therefore, the variables required for the calculation of optimal order quantity originate from:

- *Sales subsystem*: actual sale and unit cost of stock running short;
- *Storage subsystem*: unit cost of maintaining stock;
- *Procurement subsystem*: unit cost of placing a purchase order.

The remaining variables are provided by other constraint properties, as shown in Fig. 3.

SysML diagram developed in such way enables specifying a prepared set of input values and verifying the results achieved, thus comparing various variants of a target software or hardware solution. Trade-off analysis performed enables system modeler to pick an optimal solution.

5 Conclusions

SysML Parametric Diagrams have been applied in such industries as space industry, automation, military, telecommunication, biotechnology, nanotechnology, power industry, chemistry, other technical industries, healthcare or business. Their practical value in the above mentioned sectors is stimulated by the need to introduce advanced formulas, adequate for industry processes formalization. There is quite a number of cases in business and management, where advanced formulas are necessary. They are provided by experts from such domains as statistics, econometrics, operations research, theory of prognosis or financial analysis. The need for quantitative analysis creates great application fields for Parametric Diagrams. The adequacy for business systems development in respect of inventory control systems has been exemplified and confirmed in this chapter.

Acknowledgements Supporting agency: Polish Ministry of Science and Higher Education.

References

Bock C (2006) SysML and UML 2 support for activity modeling. Syst Eng 9:160–186
Booch G, Rumbaugh J, Jacobson I (2004) The UML reference manual, 2nd edn. Addison-Wesley, Boston
Dennis A, Wixom BH, Tegarden D (2005) Systems analysis and design with UML version 2.0—an object oriented approach, 2nd edn. Wiley, New York
Eriksson H, Penker M, Lyons B, Fado D (2004) UML 2 Toolkit. OMG Press, New York
Friedenthal S, Moore A, Steiner R (2008) A practical guide to SysML. OMG Press, Indianapolis
Hause M (2009) SysML hits the home straight. http://www.esemagazine.com/index.php?option=com_content&task=view&id=140&Itemid=2. Accessed 5 Aug 2009
Holt J, Perry S (2008) SysML for systems engineering. The Institution of Engineering and Technology, London

International Business Machines Corporation (IBM) (2009) IBM rational unified process—hypertext knowledge base. http://www-01.ibm.com/software/awdtools/rmc/library/. Accessed 13 Oct 2009
Object Management Group (OMG SysML) (2008) OMG Systems Modeling Language (OMG SysML). Version 1.1. http://www.sysmlforum.com/docs/specs/OMGSysML-v1.1-08-11-01.pdf. Accessed 5 Aug 2009
Object Management Group (OMG UML) (2009) OMG Unified Modeling Language (OMG UML), Superstructure. Version 2.2. http://www.omg.org/spec/UML/2.2/Superstructure/PDF/. Accessed 13 Oct 2009
Przybyłek A (2008) Separation of crosscutting concerns at the design level: an extension to the UML metamodel. In: Proceedings of international multiconference on computer science and information technology. Polish Information Processing Society Press, Mrçgowo
Technical Board International Council on Systems Engineering (2007) Systems engineering handbook. Version 3.1. http://www.incose.org/ProductsPubs/products/sehandbook.aspx. Accessed 5 Aug 2009
Weilkiens T (2007) Systems engineering with SysML/UML. Modeling, analysis, design. OMG Press, Indianapolis
Wrycza S, Marcinkowski B (2009a) Systems requirement specifications with SysML. In: Proceedings of BIR 2009: the eight international conference on perspectives in business informatics research, Academic Press, Kristianstad
Wrycza S, Marcinkowski B (2009b) The language of systems engineering—SysML. Architecture and applications. Helion, Gliwice (in Polish)
Wrycza S, Marcinkowski B, Wyrzykowski K (2005) UML 2.0 in information systems modeling. Helion, Gliwice (in Polish)

Model-Driven GUI Automation for Efficient Information Exchange Between Heterogeneous Electronic Medical Record Systems

Xuebing Yang, Yuan Miao and Yanchun Zhang

Abstract To provide high quality healthcare service, all relevant information of a patient is paramount. Most of the patient healthcare information is stored in disparate electronic medical record (EMR) systems of healthcare providers such as general practitioners (GP), specialists, hospitals etc. Integrating the existing heterogeneous EMRs for data sharing is crucial. All existing integration solutions need the numerous EMR software vendors to update their software to follow ever-changing messaging standards, provided interfaces or other criterions. The cost and efforts of the enormous amount of time required to negotiate with the large number of project participants for upgrading existing systems is extremely high. Almost exclusively, software today provides a graphical user interface (GUI) as the only method for users to access the functionality of the software. In this chapter, we propose a model for automating the procedure of operating the GUI based EMR and exchange information with them without the need of updating the existing EMR systems. The model is called parameterized GUI state model (PGUISM) which depicts the intrinsic logic of GUIs and enables the automated process of writing and reading information to and from EMR systems.

1 Introduction

High quality healthcare can be delivered only when all pertinent data on the health of a patient is available to the clinician. Integration of healthcare systems across organizational boundaries will enable all relevant information for treatment and care to be made directly available in the systems that the healthcare professionals use in their daily work [1] It is estimated that fully standardized, nationwide semantic interoperability for exchanging patient data could save $ 77.8 billion each year in redundant testing and administrative overhead in USA [2].

X. Yang (✉)
School of Engineering and Science, Victoria University, Melbourne 3011, Australia
e-mail: xuebing.yang@live.vu.edu.au

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6_65,

Due to the particularity of healthcare industry, healthcare lags behind other industries in adopting information technology by as much as 10–15 years [3], especially in integrating the isolated medical systems and making them interoperable between each other. One of the most important reasons is the heterogeneity of data sources. All the healthcare enterprises have various business processes, information levels, integration contents and demands. Most electronic medical record (EMR) systems that are currently in use are built and implemented with only local usage in mind [4]. The prime purpose of these systems is to make an isolated solution for each functional department in a hospital, GP, or organization etc., and these isolated systems can run effectively in each area [5]. The vendors' instability even leads to some obsolete systems which are almost practically impossible to be integrated to others with traditional methods. When an integration solution involves too much update of the existing systems, the cost and efforts of the enormous amount of time required to negotiate and finalize contracts with the large number of project participants is formidable. Without a proper way to avoid the updates in existing EMR systems, system integration may cost 50–100 times more than integration system development itself. Millennium bug is an example. It has estimated that globally, the expenditure was about US$ 300 billion.

To reduce the cost of integrating healthcare services and avoid updating existing EMRs, many healthcare information exchange and integration solutions leave the task of data import to and export from the EMRs to users or software vendors. The users manually operate the clinic software, navigate the system to certain states, copy the information from the software and paste it to the provided forms. After finishing the filling procedure, the information will be sent to another end. On the other side, clinic users open the received information, copy data and paste them into the EMR via its GUI and save them into the EMR.

Manual operations are unacceptably slow and easy to inject human errors into the healthcare systems. In fact, this process can be done automatically without human interfering in. Almost exclusively, the clinic software provides GUI for users to access the functionality of the software [6]. They make software easy to use by providing the user with highly visual controls that represent everyday objects such as menus, buttons, lists, and windows [7]. The GUI widgets and their properties, and values are programmatically accessible. By mimicking the process of user operations, events can be performed programmatically to the GUI widgets. To automate the process of reading from and writing to the EMR, new solution needs to be proposed for modeling the intrinsic work flow and the GUI states of the EMR.

2 Related Work

Research on GUI automation is not new. Capture and replay (CR) technique [8–11] is one of the examples. By using a CR tool, a user manually interacts with the GUI of an application. These interactions (events) are captured and recorded. The recorded information can then be replayed automatically by performing events to the GUI of the application.

Traditional CR tools provide a basic automation solution by recording mouse coordinates and user actions as scripts, which can be used later for performing routine tasks or testing the GUI based Applications (GAPs). Since these tools use mouse coordinates, the scripts break even with the slightest changes to the GUI layout.

Instead of just recording the mouse actions and coordinates, modern CR tools (e.g., Quick Test Professional(QTP), Abbot, Selenium, and Rational Functional Tester(RFT) Win runner, SilkTest, IBM Rational Robot, and Visual Test avoid this problem by capturing values of different properties of GUI objects rather than mouse coordinates. The recorded events are connected to GUI objects (widgets such as Textbox, Button, etc.) by using unique names. Unique names can be identified with collections of the values of the properties of GUI objects. When interacting with the application, the unique name will be used to get the reference to the real object in GUI and the recorded events will be performed on the designated object. The typical operations on an object are mouse click, key input, get property value, set property value etc. Objects can be classified to actionable and non-actionable objects [12]. Actionable GUI objects are those elements which can have actions performed on them which are meaningful in the context of the application. Typically, in most GUI toolkits these are buttons, menus, text boxes, etc. Non-actionable GUI objects are those which exist for layout, adornment, or other "passive" purposes such as panels and textual labels.

CR tools can only simply record user actions to a GAP, and replay it accordingly. It is very useful but still far from enough to perform true automation tasks such as GUI testing. True automation needs the knowledge of the logic or workflow of the GUIs of a GAP.

Numerous researchers have developed techniques that employ state machine models [13–19], hierarchical finite state machine(HFSM) [20, 21], graph-traversal techniques such as event flow graph (EFG), event interaction graph (EIG) [22–26] and event sequence graph (ESG) [27, 28] to generate sequences of events for creating test cases.

In finite-state machine (FSM) models and their variants, the GUI's behavior is modeled as an FSM where each input event triggers a transition in the FSM; a path in the FSM represents a test case. Since FSMs have scalability problems for large GUIs, variants such as variable finite state machines (VFSM) have been used to "collapse" some states by adding explicit conditional variables to the machine's transitions. White et al. handle the scalability problem by manually decomposing the GUI into multiple state machines called "complete interaction sequences" (CISs), each CIS models a particular manually identified GUI activity. Ana et al. [20] proposed hierarchical finite state machines (HFSMs) to deal with the state explosion problem. A HFSM is a FSM in which vertices can represent single states or groups of states (and transitions between the states of the group). These groups of states (and transitions) are themselves FSMs. The hierarchical structure of the HFSM can mimic the hierarchical structure of objects and dialogs of the GUI. Each model dialog can be seen as an independent FSM in a HFSM.

Memon et al [22–26] has conducted a series of researches for specification-based, fully automated black-box GUI testing. Unlike FSM models, instead of modeling the GUI states, event focused models such as event flow graph (EFG) and event interaction graph (EIG) are proposed. In EFG, each vertex represents an event. All events which can be executed immediately after this event are connected with directed edges from it. A path in EFG is a legal executable sequence which can be seen as a test case. EFG can be generated automatically with a tool called GUI Ripping. Test cases can be generated by traversing the EFG with certain strategy. EIG is a simplified event based model which can be generated from EFG by keeping only the system-interaction events. Belli F. et al. proposed an event sequence graph (ESG) for capturing the behavior of a variety of interactive systems that include real-time, embedded systems, and graphical user interfaces. ESGs allow a modeler to think in terms of system "events" instead of system "states". ESGs are a representation used to model the discrete interactions between an event-based system and its environment. Similar to CISs, ESG is also used to model the behavior of real-time systems. ESGs and their complements allow modeling the desirable behavior of a system in the presence of both expected and unexpected inputs as events.

The aim of these researches mentioned above is to test comprehensive systems. A large number of events and enormous value spaces of the properties in the objects have to be modeled. Because of this, no proposal has empirically demonstrated the effectiveness of the models for GUI testing [25]. To the best of our knowledge, few researches have been done for the purpose of interacting with GUI based application and exchanging information with it automatically. Other than GUI testing, to interact with the GUI, not all the events and property values of the objects are needed to be considered. This can reduce the number of the states greatly. To avoid the unnecessary prodigious cost of updating myriad existing EMRs, exchange information through their GUIs automatically for healthcare information exchange and services integration is feasible and cost effective. Our previous works have used GUI automation technology to retrieve information from clinical software in GP e-Connect [29] system and proposed a cost effective healthcare service integration solution named virtual EMR system [30] by using model driven GUI automation for autonomously interacting with the existing clinic software.

3 Parameterized GUI State Model

In this section we propose a Parameterized GUI State Model (PGUISM). This model not only is able to depict the intrinsic logic of the GUI of the EMR but also is able to automate the procedure of operating the EMR without human effort.

3.1 States of GUIs

There are many ways to define the states of a GUI application. To facilitate the GUI automation, we shall focus on GUI related state and state transitions. Thus we consider the state of a GUI application as follows: the graphical user interface of a given application is treated as a series of interfaces. Each interface can be regarded as a state. We will use this state to construct PGUISM model for GUI automation. A GUI's state is modeled as a set of opened windows and a set of objects (label, button, text, etc.) contained in each window. Hence, at a particular time t, the GUI can be represented by its constituent windows

$$
\begin{aligned}
&W = \{w_1, w_2, \ldots, w_n\} \text{ and their objects } O = \{O_1, O_2, \ldots, O_n\},\\
&\text{where } O_i = \{o(i,1), o(i,2), \ldots o(i,m_i)\},\ i = 1, 2, \ldots, n;\\
&\text{each object contains properties}\\
&P = \{P(1,1), P(1,2), \ldots, P(1,m_1),\\
&\qquad P(2,1), P(2,2), \ldots, P(2,m_2),\\
&\qquad \ldots,\\
&\qquad P(n,1), P(n,2), \ldots, P(n,m_n)\},\\
&\text{where}\\
&P(i,j) = \{p(i,j,1), p(i,j,2), \ldots, p(i,j,k_{ij})\};\\
&i = 1, 2, \ldots, n; \quad j = 1, 2, \ldots, m_i; \quad k = 1, 2, \ldots, k_{ij};\\
&\text{and their corresponding values}\\
&V(i,j) = \{v(i,j,1), v(i,j,2), \ldots, v(i,j,k_{ij})\};\\
&\text{where}\\
&i = 1, 2, \ldots, n;\\
&j = 1, 2, \ldots, m_i;\\
&k = 1, 2, \ldots, k_{ij};
\end{aligned}
$$

At a certain time t, the set of windows and their objects constitutes the state of the GUI. All the objects are organized as a forest. A GUI state then is modeled as a 4-tuple (W, O, P, V). Events $\{e_1, e_2 \ldots e_q\}$ performed on the GUI may lead to state transitions. The function notation $s_j = e_i(s_i)$ is used to denote that s_j is the state resulting from the execution of event e_i at state s_i . Such a state and transition can be considered as state machine. However, such a state machine would contain too many states and transitions that make the test automation impractical. Especially when all values of the object properties are taken into consideration, the permutation of all the states can be extremely high. To reduce the total number of states, and the number of transitions, only selected object property values are considered for distinguishing different states and only the events that lead to the state transition will be included. This greatly reduced the state number and the transitions numbers. Parameters are used to describe the context of the states.

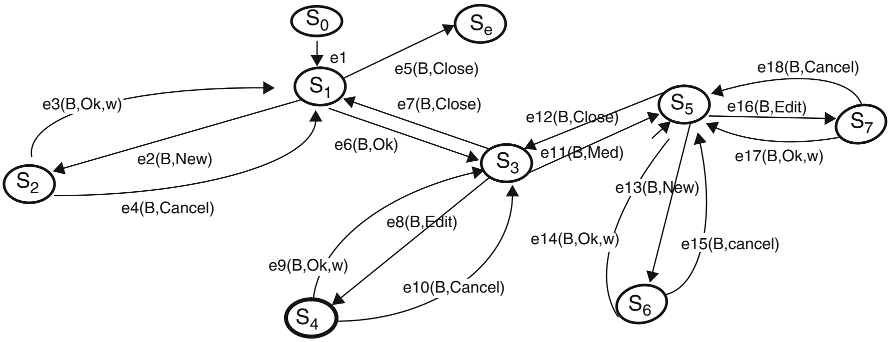


Fig. 1 A PGUISM of simple EMR

3.2 Parameterized GUI State Model

A Parameterized GUI State Model (PGUISM) is a 4-tuple (Σ, S, s_0, T, P, V), where:

- Σ is a finite, non-empty set of all possible input events of the EMR,
- S is a finite, non-empty set of GUI states of EMR,
- s_0 is an initial state, an element of S,
- T is the state-transition function set: T: $S \times \Sigma \rightarrow S$,
- P is a set of parameters, and
- V is a set of variables.

In this model, events that lead to a transition from one state to another state of an EMR constitute the input set Σ, s_0 is the first state of the EMR when it is invoked. S is composed of all possible states of the EMR. Two states are different only when the set of values of selected object properties are different from each other. The typical selected object properties are window title, checkbox, radio button and selection of a listbox etc. For each $s \in S$ is a tuple <W, E, X, Y>, where W is the object forest and E is the set of possible events in this state, $E \subseteq \sum$. $X \subseteq P$ is the set of parameters which are related to the objects in this state and $Y \subseteq V$ is the set of variables which are related to the objects in this state. For each $t \in T$, $s' = t(s, e)$, where s is the current state, s' is the next state, e is the event in s. We can also simply use <s, s′, e> to describe a transition. By performing event e on s, the state will be changed to s', $s' \neq s$. P is a set of selected parameters which are used to describe the context of current state, the selection of the parameters is manual. P is organized in hierarchical way so that the setting of value of parameters should be taken in a partial order. V is a set of variables which are used to semantically record the useful data related to certain object properties. By navigating the state under certain context and reading or writing information from or to the corresponding object properties defined in V, information can be exchanged automatically through the GUI of the EMR. Figure 1 shows the PGUSM model of simple EMR system.

3.3 Construction of PGUISM

Generating a PGUISM needs to read the widgets (objects) and perform the events on the GUI of given EMR. We have built a set of fundamental tools which read the widgets (objects), check states and perform events on the GUI. ReadState reads the current state of a given EMR, Existing(s, S) tells whether s is contained in S according to certain criteria, IsDifferent(s_1, s_2) checks whether s_1 and s_2 are different states, and GetEquivalentState(s, S) returns the state in S which is equivalent to s according to certain criteria. The procedure of the construction is maneuvered by a user and the events taken by the user are hooked and recorded. Figure 2 shows the mechanism of the construction of PGUSM, and Algorithm 1 generates the PGUISM for a given EMR.

Algorithm 1 GenPGUISM

```
1.  Σ ← Ø , S←Ø, T ← Ø , P ← Ø,  V ← Ø ;
2.  Start the EMR;
3.  Handle =GetHandle(APPname) //use API to find the handle
4.  If (handle==null) goto 23.
5.  Sc= new State()
6.  Sc.W= ReadState(handle) //read objects and their sub objects recursively and generate a
    object tree of current GUI.
7.  S.X←  Create and Map objects to variables if needed.(manually)
8.  S.Y← Create and  Map objects to parameters if needed.(manually)
9.  P←P ∪ S.X; V← V ∪ S.Y
10. S← S ∪ {sc}; s0 ← sc;
11. Wait for new event:  lastevent = CheckEvent()
12. If appexited(handle) goto 23
13. s1=new state()
14. s1.W=ReadState(handle)
15. if(!IsDifferent(sc,s1)) goto 11.
16. if Existing(s1, S)   s1= GetEquivalentState(s1,S);
17. t=new transition(<sc,s1,lastevent>)
18. Σ ← Σ ∪ {lastevent}; Sc.E = sc.E ∪ {lastevent};
19. T ← T ∪ {t};
20. S ← S ∪ {s1};
21. Sc=s1;
22. Goto 11;
23. End
```

3.4 Automation of Information Exchange Using PGUISM

Once the PGUISM is created, it can be used to operate the EMR system automatically. Figure 3 shows the mechanism of automating the interaction with EMR by using PGUISM. Given a task, e.g., retrieving patient information, a request info variable list will be sent to the Message exchanger. The Objectives finder will search the states which include the requested information variables in the PGUISM. After all needed states are found, the GUI Navigator will navigate the EMR from current state to each of the found state and perform actions either reading information from the

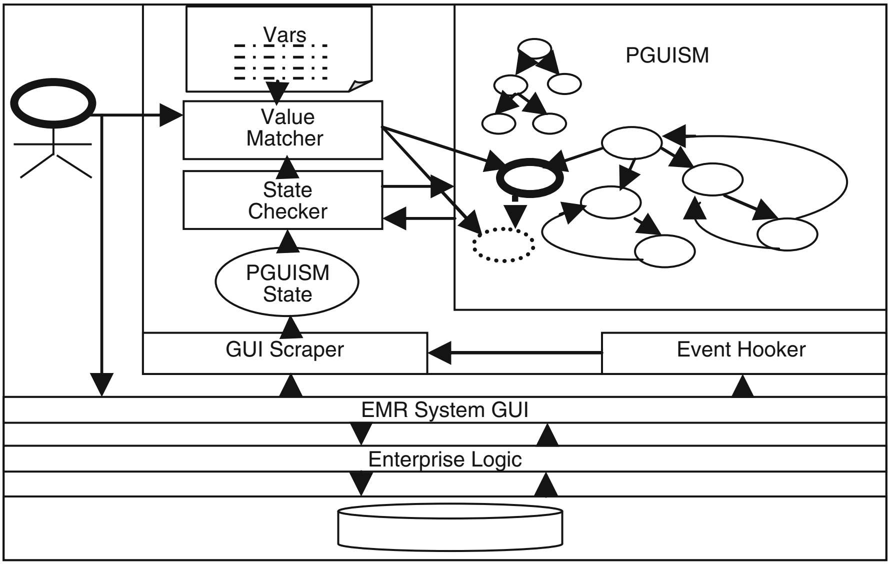


Fig. 2 Mechanism of constructing a PGUISM for an EMR

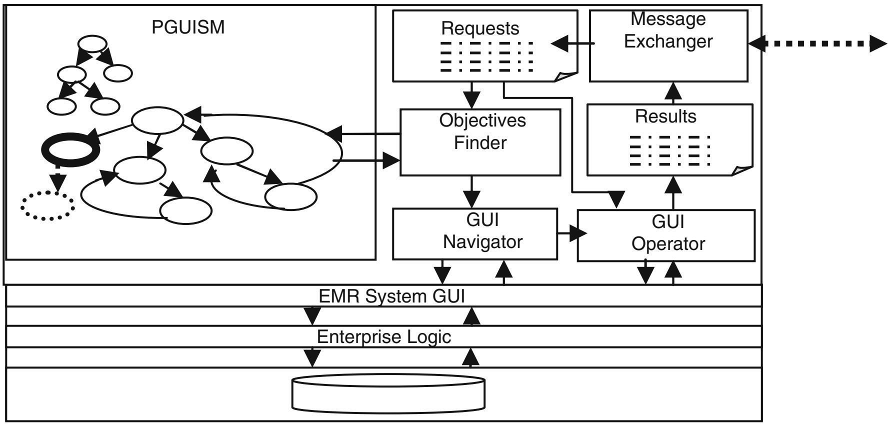


Fig. 3 Mechanism of automating an EMR system

real-time object property value to the related variables or writing information from the variables to the real-time object properties values. To navigate to a given state, a path from current state to the given state can be figured out by graph searching. With the path information, GUI operator will perform the events recorded in the transitions automatically in sequence to reach the destination state. After all related states are visited, results which include either the required information or writing reports will be generated and will be sent back through Message Exchanger. Algorithm 2 shows the procedure of executing a task for reading/writing information from/to the GUIs of an EMR system.

Algorithm 2 ExecuteTask (variableList, read/write)

```
G←OpenPGUISM(fileName)
S=SearchStates(G, variableList) //S is a set of states which include defined vari-
ables in variableList.
Foreach s ∈ S {
  P=FindPathFromCurrentTo(s);    //Graph search
  For each p ∈ P {
   sc=readState();              //read the current state
    PerformEvent(p.e, sc);      //perform the event to currentGUI
    Wait();                      // for next state appear
  }
  S'=ReadState();
  If(IsDifferent(s',s)) Error.
  If(read) results = results ∪ ReadInformationFrom(s',variableList)
  Else      results= results ∪  WriteInformationTo(variableList, s')
 }
 Return results
```

4 Empirical Studies

To check the effectiveness and feasibility of PGUISM, we implemented PGUISM in C# and use UIAutomation library in Visual Studio.net 2008 to read widgets information from an application. Two of the most popular clinic software in Australia, Medical Director (MD) and Best Practice (BP) are selected for evaluation. Our purpose is to create PGUISM for each of the software and create a mechanism to exchange information between these two clinic software. The information we selected for exchanging is the content required by Victoria Standard Referral Format (VSRF), a referral standard in Victoria for exchanging information between general practitioners, specialists, hospitals and other healthcare service providers.

To control the number of states, only the states which contain the required information are considered. Other states that are on the way from the initial state s_0 to the above states are also collected for navigating purpose. Table 1 are the number of states, transitions and other information in PGUISM for both MD and BP.

From Table 1 we can see that the state number is very small. If we take into consideration of all the states of the software, the number is much larger than this. Our model is to exchange referral information, and only the states which contain the related information are selected.

Transitions are directed edges that start from one state and end in another state. In each edge, the corresponding actions (events) are recorded for automatic events performing and states navigating. There are always multiple paths from one state to another state in real system, but for data exchange automation, only the efficient transition actions are selected. This greatly reduced the number of transitions that are used in GUI test automation.

Table 1 PGUISM model information of MD and BP

Software	States	Transitions	Widgets	Parameters	Variables	File size (kb)
MD	24	53	2,237	7	112	418
BP	30	58	4,796	7	53	752

Table 2 Comparison of tasks execution time

Software	Manual/ automatic	Read and create referral (s)	Import referral (s)
MD	Manual	960	360
	Automatic	15	20
BP	Manual	1,020	480
	Automatic	18	19

Widgets number is the total number of all states. In BP, because the information is shown in more detailed way, the total number is much larger than the number in MD. When automating the tasks, it takes more time to load all the widgets and retrieve properties values in BP than it does in MD.

Parameters are used for depicting the context. Before performing each task, the values of the parameters are supposed to be filled in advance. In both MD and BP, we use the same parameters. Variables are used for reading information from the GUI and writing information to the GUI. Because the two software show information differently, the numbers of variables used in the two models are different.

To test the efficiency of the automation, we performed tasks on both MD and BP. The Task 1 is to read current patient information and create a new referral from MD and BP respectively. Task 2 is to import the received referral into MD or BP respectively. We perform the same tasks manually as well. Table 2 shows the results of the test. From Table 2 we can see that automatic process is much faster than manual process. Automatic process can also avoid producing entry errors.

5 Conclusion

This chapter has introduced a model named parameterized GUI state model (PGUISM) for depicting the intrinsic logic and work flow of graphic user interfaces (GUIs) of EMRs. With the model, clinic information can be exchanged with the EMR automatically without involving human operations. This approach provides an economic solution to integrating heterogeneous healthcare services. No updates to existing EMRs are needed. This saves the formidable cost of updating, testing and deploying the existing heterogeneous healthcare software which are already used among general practitioners, specialists, hospitals and other healthcare providers.

Empirical study shows that the model driven GUI automation can run smoothly and effectively on two most popular clinic software in Australia. The automated process for creating electronic referral for the current patient from both of the two software and importing referral information into the software are much faster than manual process. The data collected by the automated process are structured and manual errors are avoided.

References

1. Dalsgaard E, Kjelstrom K, Riis J (2008) A federation of web services for Danish health care ID trust, 4–6 Mar 2008, Gaithersburg
2. Walker J, Pan E, Johnston D, Adler-Milstein J, Bates DW, Middleton B (2005) The value of health care information exchange and interoperability. Health Aff (Millwood) 2005: 10–18 (19 Jan 2005)
3. Goldschmidt P (2005) HIT and MIS: implications of health information technology and medical information systems. Commun ACM 48(10):69–74 (Oct)
4. van der Linden H, Kalra D, Hasman A, Talmon J (2009) Inter-organizational future proof EHR systems: a review of the security and privacy related issues. Internet J Med Inf 78:141–160
5. Wang W, Wang M, Zhu S (2005) Healthcare information system integration: a service oriented approach. Services systems and services management, 2005. In: Proceedings of ICSSSM '05, vol 2. IEEE Computer Society, Washington, pp 1475–1480
6. Brooks P, Robinson B, Memon AM (2009) An initial characterization of industrial graphical user interface systems. In: ICST 2009: proceedings of the 2nd IEEE international conference on software testing, verification and validation. IEEE Computer Society, Washington
7. Memon AM, Banerjee I, Nagarajan A (2003) GUI ripping: reverse engineering of graphical user interfaces for testing. In: Proceedings of The 10th working conference on reverse engineering, Victoria, Nov
8. Anderson J, Bache G (2004) The video store revisited yet again: adventures in GUI acceptance testing. In: Proceedings of the 5th international conference on extreme programming and agile processes in software engineering, LNCS 3092. Springer, Berlin, pp 1–10
9. Dutta S (2003) Abbot—a friendly JUnit extension for GUI testing. Java Dev J 8:8–12
10. Lowell C, Stell-Smith J (2003) Abbot—a friendly JUnit extension for GUI testing. LNCS 2675:331–333
11. Finsterwalder M (2001) Automating acceptance tests for GUI applications in an extreme programming environment. In: Proceedings of the 2nd international conference on extreme programming and flexible processes in software engineering, Sardinia, Italy, pp 20–23
12. McMaster S, Memon AM (2009) An extensible heuristic-based framework for GUI test case maintenance. ICSTW, IEEE international conference on software testing, verification, and validation workshops, pp 251–254
13. Clarke JM (1998) Automated test generation from a behavioral model. In: Proceedings of Pacific northwest software quality conference. Pnsqc/Pacific Agenda, Portland
14. Chow TS (1978) Testing software design modeled by finite-state machines. IEEE Trans Softw Eng SE 4(3):178–187
15. Esmelioglu S, Apfelbaum L (1997) Automated test generation, execution, and reporting. In: Proceedings of Pacific northwest software quality conference. Pnsqc/Pacific Agenda, Portland, pp 127–142
16. Bernhard PJ (1994) A reduced test suite for protocol conformance testing. ACM Trans Softw Eng Methodol 3(3):201–220 (July)
17. Shehady RK, Siewiorek DP (1997) A method to automate user interface testing using variable finite state machines. In: Proceedings of the 27th annual international symposium on fault-tolerant computing (FTCS'97). IEEE Press, Washington, Brussels, Tokyo, pp 80–88
18. White L, Almezen H (2000). Generating test cases for GUI responsibilities using complete interaction sequences. In: ISSRE '00: proceedings of the 11th international symposium on software reliability engineering (ISSRE'00). IEEE Computer Society, Washington, DC, p 110

19. Offutt AJ, Hayes JH (1996) A semantic model of program faults. In: ISSTA'96: proceedings of the 1996 ACM SIGSOFT international symposium on software testing and analysis. ACM Press, New York, pp 195–200
20. Paiva ACR, Tillmann N, Faria JCP, Vidal RFAM (2005) Modeling and testing hierarchical GUIs. In: Proceedings of the 12th international workshop on abstract state machines, Paris, France, pp 153–165
21. Paiva ACR (2009) Automated GUI testing in informàtica. In: XIII Convención Y Feria Internacional Informàtica, Publicado por Lesman on Sun
22. Memon AM, Xie Q (2005) Studying the fault-detection effectiveness of GUI test cases for rapidly evolving software. IEEE Trans Softw Eng 31:884–896 0(Oct)
23. Xie Q, Memon AM (2005) Rapid "crash testing" for continuously evolving GUI-based software applications. In: ICSM '05: proceedings of the 21st IEEE international conference on software maintenance (ICSM'05). IEEE Computer Society, Washington, pp 473–482
24. Xie Q, Memon AM (2006) Automated model-based testing of community-driven open source GUI applications. In: ICSM '06: proceedings of the 22nd IEEE international conference on software maintenance. IEEE Computer Society, Washington, pp 145–154
25. Xie Q, Memon AM (2008) Using a pilot study to derive a GUI model for automated testing. ACM transactions on software engineering and methodology. ACM Press, New York
26. Memon AM (2009) An event-flow model to test EDS. In: Belini EA (ed) Software engineering and development. Nova Science Publishers, Hauppauge, New York
27. Zhu H, Wong WE, Belli F (2008) Advancing test automation technology to meet the challenges of model-driven software development: report on the 3rd workshop on automation of software test. ICSE, Leipzig, Germany
28. Belli F (2001) Finite-state testing and analysis of graphical user interfaces, ISSRE. Paderborn University, Germany
29. Yang X, Miao Y, Zhang Y (2009) GP eConnect: extends e-referrals exchange to healthcare providers' collaborations. IADIS eHealth 2009. Algarve, Portugal, 21–23 June
30. Yang X, Miao Y (2011) Distributed agent based interoperable virtual EMR system for healthcare system integration, J Med Sys 35(3):309–319

Index

J. Pokorny et al. (eds.), *Information Systems Development*,
DOI 10.1007/978-1-4419-9790-6, © Springer Science+Business Media, LLC 2011