

Information Systems Development

William Wei Song · Shenghua Xu ·
Changxuan Wan · Yuansheng Zhong ·
Wita Wojtkowski · Gregory Wojtkowski ·
Henry Linger
Editors

Information Systems Development

Asian Experiences

Foreword by William Wei Song and Shenghua Xu

 Springer

Editors

William Wei Song
Durham University
Durham DH1 3LE, UK
w.w.song@durham.ac.uk

Shenghua Xu
School of Information Technology
Jiangxi University of Finance
and Economics
Nanchang Jiangxi 330013, China
shenghua.xu@jxufe.edu.cn

Changxuan Wan
School of Information Management
Jiangxi University of Finance
and Economics
Nanchang Jiangxi 330013, China
changxuan.wan@jxufe.edu.cn

Yuansheng Zhong
UFIDA Software Technology College
Jiangxi University of Finance
and Economics
Nanchang Jiangxi 330013, China
zhong_ys@hotmail.com

Wita Wojtkowski
Department of Information Technology
and Supply Chain Management
Boise State University
Boise, ID 83725-1615, USA
riswoj2t@cobfac.boise.state.edu

Gregory Wojtkowski
Department of Information Technology
and Supply Chain Management
Boise State University
Boise, ID 83725-1615, USA
gwojtkow@boisestate.edu

Henry Linger
Monash University
Clayton, VIC 3800, Australia
henry.linger@infotech.monash.edu.au

ISBN 978-1-4419-7205-7
DOI 10.1007/978-1-4419-7355-9
Springer New York Dordrecht Heidelberg London

e-ISBN 978-1-4419-7355-9

Library of Congress Control Number: 2010938865

© Springer Science+Business Media, LLC 2011

All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, LLC, 233 Spring Street, New York, NY 10013, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

Printed on acid-free paper

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

Foreword

(Messages from Program Co-Chairs)

The 18th International Conference on Information Systems Development (ISD2009) is held in Nanchang, China, from September 16 to 19, 2009.

In keeping with the traditions of the conference, ISD2009, which is organized by Jiangxi University of Finance and Economics, aims to provide an international platform for the discussion and exchange of new research ideas on information system development and related among researchers, developers, and users from around the world. The theme of the conference is “Asian Experience”.

This year, we received 89 papers from more than 20 counties and regions. These papers have been rigorously reviewed by the Program Committee members, each paper being reviewed by least three reviewers. Finally, 54 papers have been accepted for presentation at the conference, setting an acceptance rate of 61%. The accepted papers are organized for presentation in 14 sessions spanning 9 tracks: “Enterprise Systems – A Challenge for Future Research and Improved Practice”, “Business Systems Analysis”, “IS/IT Project Management”, “Data and Information Systems Model”, “Human–Computer Interaction in ISD”, “Information Systems for Service Marketing”, “Development of Information Systems for Creativity and Innovation”, “Model-Driven Engineering in ISD”, “Legal and Administrative Aspects of Information Systems Development”, “Information Systems Engineering and Management”, and “Agile and High-Speed Systems Development”.

The conference is privileged to host five keynote speeches delivered by Guoqing Chen (Tsinghua University, China), Jian Ma (City University of Hong Kong, China), Zhi Jin (Peking University, China), Chengqi Zhang (University of Technology, Australia), and Deli Yang (Dalian University of Technology, China). The speeches provide an insight into a variety of research topics including e-Business, Internet-based software development, data mining, and electronic commerce and present challenges on related research issues.

The conference would not have been a success without the hard work of many individuals, including the general co-chairs Guoqin Chen and Qiao Wang, the organizing co-chairs Changxuan Wan, Guoqiong Liao, Bo Shen, Shumei Liao, and Yuansheng Zhong, and others in the organizing committees, thanks for putting

an exceptional programme. We are also grateful for all the PC members and the external reviewers for setting up a high standard for our conference. We would also like to thank the keynote speakers, the track chairs, the authors in particular who have contributed to the success of our conference. We would like to thank our sponsor NSFC. Last but not least, a big “thank you” to Jiangxi University of Finance and Economics for their great effort in organizing the event.

Durham, UK
Nanchang, China

William Wei Song
Shenghua Xu
Program Co-chairs

Preface

(Messages from the General Co-Chairs)

On behalf of the organizing committee, we would like to welcome you to the 18th International Conference on Information Systems Development (ISD2009). The purpose of this conference is to provide an international forum for technical discussion and exchange of ideas in information systems area among the researchers, developers, and users from academia, business, and industry.

This is the first time that this conference is held in China. We are most privileged to be hosting this conference. Nanchang is an ancient city full of historical and cultural heritages dating back more than 2,200 years. As the capital city of Jiangxi Province, it is an excellent location for a forum of academic and professional communications and exchange of research ideas, with a side helping of meaningful entertainment and cultural immersion.

In spite of the negative impact of the financial recession starting from 2008 and the prevalence of A/H1N1 Flu, the conference still received many papers from more than 20 counties and regions this year.

This year's conference continues the ISD conferences tradition with an excellent program, consisting of 5 keynote speeches and 54 research paper presentations. The accepted papers have been peer reviewed strictly by three or more experts and selected carefully from 89 submissions across Australia, China (including Hong Kong), Croatia, Czech Republic, Denmark, Finland, Germany, Greece, Hong Kong, India, Ireland, Latvia, Lithuania, the Netherlands, Poland, Spain, Sweden, the UK, and the USA.

We wish to express our gratitude to the programme committee members, external reviewers, track chairs, and all the authors in particular for their excellent contributions of their work.

We would like to thank the International Advisory Committee, Gregory Wojtkowski, Wita Wojtkowski, and Henry Linger for their guidance. We also appreciate the work by the programme co-chairs, William Wei Song and Shenghua Xu; the organizing co-chairs, Changxuan Wan and Guoqiong Liao; the organizing vice-chairs, Bo Shen, Shumei Liao, and Yuansheng Zhong; the finance and registration chair, Jucheng Yang; the local arrangements chair, Shihua Luo; the publication

chair, Xiaofeng Du; and the publicity co-chairs, Joe Geldart and Juling Ding, who have done a great job putting together an excellent technical program.

Finally, but none the less, thanks to NSF China, Jiangxi Computer Society, and Jiangxi University of Finance and Economics for their sponsorship and support.

We hope that all of you will find the technical programme of ISD2009 to be interesting and beneficial to your research. We also hope that you enjoy your stay in Nanchang, with time to visit the plethora of historic and scenic locations, such as Tengwang Pavilion, Lushan Mountain, Shanzong Mountain, and leave with a memorable experience of China.

Beijing, China
Nanchang, China

Guoqing Chen
Qiao Wang

Contents

Part I	Enterprise Systems – A Challenge for Future Research and Improved Practice	
CbSSDF and OWL-S: A Scenario-Based Solution Analysis and Comparison		3
Xiaofeng Du, William Wei Song, and Malcom Munro		
Enterprise Systems in a Service Science Context		15
Anders G. Nilsson		
A Petri Net-Based Software Process Model for Developing Process-Oriented Information Systems		27
Yu Li and Andreas Oberweis		
Modern Enterprise Systems as Enablers of Agile Development		41
Odd Fredriksson and Lennart Ljung		
Patterns-Based IS Change Management in SMEs		55
Janis Makna and Marite Kirikova		
Applying Use Cases to Describe the Role of Standards in e-Health Information Systems		67
Emma Chávez, Gavin Finnie, and Padmanabhan Krishnan		
Discussion on Development Trend of Chinese Enterprises Information System		77
Xiao-hong Gan		
Asymmetrical Effects of Using Positive and Negative Examples on Object Modeling		85
Narasimha Bolloju, Christoph Schneider, and Doug Vogel		
Part II	IS/IT Project Management	
A Social Contract for University–Industry Collaboration: A Case of Project-Based Learning Environment		99
Tero Vartiainen		

Replacement of the Project Manager Reflected Through Activity Theory and Work-System Theory	111
Tero Vartiainen, Heli Aramo-Immonen, Jari Jussila, Maritta Pirhonen, and Kirsi Liikamaa	
Integrating Environmental and Information Systems Management: An Enterprise Architecture Approach	123
Ovidiu Noran	
Effective Monitoring and Control of Outsourced Software Development Projects	135
Laura Ponisio and Peter Vruggink	
Classification of Software Projects' Complexity	149
P. Fitsilis, A. Kameas, and L. Anthopoulos	
Application of Project Portfolio Management	161
Malgorzata Pankowska	
 Part III Human-Computer Interaction and Knowledge Management	
Towards a Cost-Effective Evaluation Approach for Web Portal Interfaces	175
Andrina Granić, Ivica Mitrović, and Nikola Marangunić	
IT Knowledge Requirements Identification in Organizational Networks: Cooperation Between Industrial Organizations and Universities	187
Peteris Rudzajs and Marite Kirikova	
A Knowledge Tree Model and Its Application for Continuous Management Improvement	201
Yun Lu, Zhen-Qiang Bao, Yu-Qin Zhao, Yan Wang, and Gui-Jun Wang	
On the Development of a User-Defined Quality Measurement Tool for XML Documents	213
Eric Pardede and Tejasvi Gaur	
The Paradox of “Structured” Methods for Software Requirements Management: A Case Study of an e-Government Development Project	223
Kieran Conboy and Michael Lang	
The Research for Knowledge Management System of Virtual Enterprise Based on Multi-agent	233
Yang Bo and Shenghua Xu	

Part IV Model-Driven Engineering in ISD

Problem-Solving Methods in Agent-Oriented Software Engineering . . .	243
Paul Bogg, Ghassan Beydoun, and Graham Low	
MORPHEUS: A Supporting Tool for MDD	255
Elena Navarro, Abel Gómez, Patricio Letelier, and Isidro Ramos	
Towards a Model-Driven Approach to Information System Evolution . .	269
Mohammed Aboulsamh and Jim Davies	
Open Design Architecture for Round Trip Engineering	281
Miroslav Beličák, Jaroslav Pokorný, and Karel Richta	
Quality Issues on Model-Driven Web Engineering Methodologies	295
F. J. Domínguez-Mayo, M.J. Escalona, and M. Mejías	
Measuring the Quality of Model-Driven Projects with NDT-Quality . . .	307
M.J. Escalona, J.J. Gutiérrez, M. Pérez-Pérez, A. Molina, E. Martínez-Force, and F.J. Domínguez-Mayo	
Aligning Business Motivations in a Services Computing Design	319
T. Roach, G. Low, and J. D'Ambra	

Part V Information Systems for Service Marketing and e-Businesses

CRank: A Credit Assessment Model in C2C e-Commerce	333
Zhiqiang Zhang, Xiaoqin Xie, Haiwei Pan, and Qilong Han	
Towards Agent-Oriented Approach to a Call Management System . . .	345
Amir Nabil Ashamalla, Ghassan Beydoun, and Graham Low	
Design and Research on e-Business Platform Based on Agent	357
L.Z. Li and L.X. Li	
Research of B2B e-Business Application and Development Technology Based on SOA	367
Li Liang Xian	
Dynamic Inventory Management with Demand Information Updating . .	377
Jian Liu and Chunlin Luo	
Analysis of Market Opportunities for Chinese Private Express Delivery Industry	383
Changbing Jiang, Lijun Bai, and Xiaoqing Tong	

Part VI Development of Information Systems for Creativity and Innovation

Explaining Change Paths of Systems and Software Development Practices	399
Kari Smolander, Even Åby Larsen, and Tero Päivärinta	

Designing a Study for Evaluating User Feedback on Pre-design Models	411
Jürgen Vöhringer, Peter Bellström, Doris Gälle, and Christian Kop	
Study on the Method of the Technology Forecasting Based on Conjoint Analysis	425
Jing-yi Miao, Cheng-yu Liu, and Zhen-hua Sun	
Towards a New Concept for Supporting Needy Children in Developing Countries – ICT Centres Integrated with Social Rehabilitation	437
Anders G. Nilsson and Thérèse H. Nilsson	
An Investigation of Agility Issues in Scrum Teams Using Agility Indicators	449
Minna Pikkarainen and Xiaofeng Wang	
The Influence of Short Project Timeframes on Web Development Practices: A Field Study	461
Michael Lang	
On Weights Determination in Ideal Point Multiattribute Decision-Making Model	475
Xin-chang Wang and Xin-ying Xiao	
 Part VII Information Systems Engineering and Management	
An Approach for Prioritizing Agile Practices for Adaptation	485
Gytenis Mikulenas and Kestutis Kapocius	
Effects of Early User-Testing on Software Quality – Experiences from a Case Study	499
John Sören Pettersson and Jenny Nilsson	
Development of Watch Schedule Using Rules Approach	511
Darius Jurkevicius and Olegas Vasilecas	
Priority-Based Constraint Management in Software Process Instantiation	523
Peter Killisperger, Markus Stumptner, Georg Peters, and Thomas Stückl	
Adopting Quality Assurance Technology in Customer–Vendor Relationships: A Case Study of How Interorganizational Relationships Influence the Process	535
Lise Tordrup Heeager and Gitte Tjørnehøj	
A Framework for Decomposition and Analysis of Agile Methodologies During Their Adaptation	547
Gytenis Mikulenas and Kestutis Kapocius	

**The Methodology Evaluation System Can Support Software
Process Innovation** 561
Alena Buchalcevova

Index 573

The 18th International Conference on Information Systems Development (ISD2009)

**Nanchang, China
September 16–19, 2009**

Organizer

School of Information Technology
Jiangxi University of Finance & Economics
Tel: +86-791-3983891
Fax: +86-791-3983891
Email: isd2009.jxufe@gmail.com isd2009@jxufe.edu.cn

Conference Organization

ISD2009 Conference Committee

GENERAL CO-CHAIRS

Guoqin Chen, Tsinghua University, China

Qiao Wang, Jiangxi University of Finance & Economics, China

PROGRAMME CO-CHAIRS

William Wei Song, University of Durham, UK

Shenghua Xu, Jiangxi University of Finance & Economics, China

INTERNATIONAL ADVISORY COMMITTEE

Gregory Wojtkowski, Boise State University, USA

Wita Wojtkowski, Boise State University, USA

Henry Linger, Monash University, Australia

ORGANIZATION CO-CHAIRS

Changxuan Wan, Jiangxi University of Finance & Economics, China

Guoqiong Liao, Jiangxi University of Finance & Economics, China

ORGANIZATION VICE-CHAIRS

Bo Shen, Jiangxi University of Finance & Economics, China

Shumei Liao, Jiangxi University of Finance & Economics, China

Yuansheng Zhong, Jiangxi University of Finance & Economics, China

FINANCE AND REGISTRATION CHAIR

Jucheng Yang, Jiangxi University of Finance & Economics, China

LOCAL ARRANGEMENTS CHAIR

Shihua Luo, Jiangxi University of Finance & Economics, China

PUBLICATION CO-CHAIRS

Bo Shen, Jiangxi University of Finance & Economics, China

Xiaofeng Du, University of Durham, UK

PUBLICITY CO-CHAIRS

Joe Geldart, University of Durham, UK

Juling Ding, Jiangxi University of Finance & Economics, China

TRACK CHAIRS

Rónán Kennedy, National University of Ireland, Galway, Ireland

Anders G. Nilsson, Karlstad University, Sweden

Odd Fredriksson, Karlstad University, Sweden

Maria Jose Escalona, University of Sevilla, Spain

Malgorzata Pankowska, University of Economics, Katowice, Poland

Yang Li, BT Innovate

Aimilia Tzanavari, University of Nicosia, Cyprus

Andrina Granic, University of Split, Croatia

Michael Lang, National University of Ireland, Galway

Kieran Conboy, National University of Ireland, Galway

PC MEMBERS

Abrahamo Silvia, Technical University of Valencia

Backlund Per, University of Skovde, Sweden

Barn Balbir, Thames Valley University, UK
Bashroush Rabih, Queen's University Belfast, UK
Bečejski-Vujaklija Dragana, Faculty Of Organizational Sciences, Serbia
Bellström Peter, Karlstad University, Sweden
Bezzazi El-Hassan, University de Lille 2, France
Borzovs Juris, Information Technology Institute, Latvia
Buchalceková Alena, University of Economics, Chzech Republic
Burdescu Dumitru, University of Craiova, Romania
Burstein Frada, Monash University, Australia
Carlsson Sven, Lund University, Sweden
Ceccez-Kecmanovic Dubravka, University of New South Wales, Australia
Charalabidis Yannis, National Technical University of Athens
Chen Yu, Renmin University of China, China
Coady Jenny, Heriot-Watt University, UK
Connor Andy, Auckland University of Technology, New Zealand
Cuzzocrea Alfredo, ICAR Institute and DEIS Department, University of Calabria, Italy
Du Xiaofeng, Durham University, UK
Eessaar Erki, Tallinn University of Technology, Estonia
Exton Chris, University of Limerick, Ireland
Finnie Gavin, Bond University, Australia
Fitsilis Panos, TEI Larisas
Forsell Marko, SESCO Technologies, Finland
Fredriksson Odd, Karlstad University, Sweden
Gadatsch Andreas, FH Bonn-Rhein-Sieg, Germany
Granic Andrina, University of Split, Croatia
Gutiérrez Rodríguez Javier Jesús, University Of Sevilla, Spain
Hawryszkiewicz Igor, University of Technology Sydney, Australia
Hevner Alan, University of South Florida, USA
Huisman Magda, North-West University, South Africa

Ivanov Sergey, George Washington University, USA
Ivanovic Mirjana, University of Novi Sad, Serbia
Jaccheri Letizia, Norwegian University of Science and Technology, Norway
Janiesch Christian, SAP Australia Pty Ltd, Australia
Johansson Björn, Copenhagen Business School, Denmark
Juvonen Pasi, South Carelia Polytechnic, Finland
Kaschek Roland, Massey University, New Zealand
Kautz Karlheinz, Copenhagen Business School, Denmark
Kirikova Marite, Riga Technical University, Latvia
Kop Christian, Alpen-Adria Universitaet Klagenfurt, Austria
Krogstie John, Norwegian University of Science and Technology, Norway
Kruithof Gert, TNO Information and Communication Techno, Netherlands
Kuras Marian, Cracow Academy of Economics, Poland
Kuznetsov Sergei, Institute for System Programming of Russia, Russia
Lam Vitus, The University of Hong Kong, Hong Kong
Lang Michael, National University of Ireland, Galway
Leppänen Mauri, University of Jyväskylä, Finland
Linger Henry, Monash University, Australia
Liu Lu, Beijing University of Aeronautics & Astronautics, China
Low Graham, University of New South Wales, Australia
Mallinson Brenda, Rhodes University, South Africa
Manolopoulos Yannis, Aristotle University, Greece
Marghescu Dorina, Turku Centre for Computer Science/Abo Akademi University, Finland
Maria Jose Escalona, University of Sevilla, Spain
Mathiassen Lars, Georgia State University, US
Mavromoustakos Stefanos, European University, Cyprus
Medve Anna, University of Pannonia Veszprem, Hungary
Melin Ulf, Linköping University, Sweden
Metais Elisabeth, CNAM, France

Middleton Peter, Queen's University Belfast, UK
Molloy Owen, National University of Ireland, Sweden
Moreton Robert, University of Wolverhampton, UK
Mouratidis Haris, University of East London, UK
Munro Malcolm, Durham University, UK
Nachev Anatoli, National University of Ireland, Ireland
Nemuraite Lina, Kaunas Technical University, Lithuania
Niehaves Björn, Institut für Wirtschaftsinformatik, Germ
Noran Ovidiu, Griffith University, Australia
Nørbjerg Jacob, Copenhagen Business School, Denmark
Ovaska Päivi, South Carelia Polytechnic, Finland
Owen Jill, University College Canberra, Australia
Pade Caroline, Rhodes University, South Africa
Pastor Oscar, University of Valencia, Spain
Persson Anne, University of Skövde, Sweden
Pirhonen Maritta, University of Jyväskylä, Finland
Pirotte Alain, University of Louvain, Belgium
Plantak Vukovac Dijana, University of Zagreb, Croatia
Plata-Przechlewski Tomasz, Gdansk University, Poland
Pokorny Jaroslav, Charles University, Czech Republic
Przybylek Adam, Gdansk University, Poland
Rachev Boris, Technical University of Varna, Bulgaria
Ramos Salavert Isidro, University Politecnica de Valencia, Spain
Richta Karel, Czech Technical University, Czech Republic
Robal Tarmo, Tallinn University of Technology, Estonia
Rouibah Kamel, Kuwait University
Schiopiu Burlea Adriana, University of Craiova, Romania
Shoval Peretz, Ben-Gurion University, Israel
Stamelos Ioannis, Aristotle University, Greece
Steen Odd, Lund University, Sweden

Strahonja Vjeran, University of Zagreb, Croatia

Strasunskas Darijus, NTNU, Norway

Sukovskis Uldis, Riga Technical University, Latvia

Sundgren Bo, Mid Sweden University, Sweden

Tikk Domonkos, Budapest University of Technology and Economics, Hungary

Traxler John, University of Wolverhampton, UK

Vartiainen Tero, Turku School of Economics, Finland

Vavpotič Damjan, University of Ljubljana, Slovenia

Vintere Anna, Latvia University of Agriculture

Vorisek Jiri, Prague University of Economics, Czech Republic

Vossen Gottfried, University of Münster, Germany

Wang Hongbing, Southeast University, China

Wang Kan-liang, Xi'an Jiao Tong University, China

Wastell Dave, University of Salford, UK

Welch Christine, University of Portsmouth, UK

Wrycza Stanislaw, University of Gdansk, Poland

Xavier Costa Heitor Augustus, University Federal de Lavras, Brazil

Zarri Gian Piero, University Paris 4/Sorbonne, France

Zurada Jozef, University of Louisville, USA

Contributors

Mohammed Aboulsamh Oxford University Computing Laboratory, Oxford, UK,
mohammed.aboulsamh@comlab.ox.ac.uk

L. Anthopoulos Technological Education Institute of Larissa, 41110 Larissa,
Greece, anthopoulos@teilar.gr

Heli Aramo-Immonen Department of Industrial Management and Engineering,
Tampere University of Technology, Pori, Finland

Amir Nabil Ashamalla School of Information Systems and Technology,
University of Wollongong, Wollongong, NSW, Australia, ana462@uow.edu.au

Lijun Bai College of Information Management, Zhejiang Gongshang University,
Hangzhou, China, blj@mail.zjgsu.edu.cn

Zhen-qiang Bao Information Engineering College, Yangzhou University, Jiangsu
225009, China

Miroslav Beličák Department of Software Engineering, Faculty of Mathematics
and Physics, Charles University in Prague, Prague, Czech Republic,
miroslav.belicak@mff.cuni.cz

Peter Bellström Department of Information Systems, Karlstad University,
Karlstad, Sweden, peter.bellstrom@kau.se

Ghassan Beydoun School of Information Systems and Technology, University of
Wollongong, Wollongong, NSW, Australia, beydoun@uow.edu.au

Yang Bo School of Information Management, Jiangxi University of Finance &
Economics, Nanchang, China, yb_lh@126.com

Paul Bogg University of New South Wales, Sydney, NSW, Australia,
paulb@unsw.edu.au

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

Alena Buchalceva Department of Information Technologies, University of Economics, Prague, Prague, Czech Republic, buchalc@vse.cz

Emma Chávez School of Information Technology, Bond University, Robina, Gold Coast, QLD, Australia; Departamento de Ing. Informática, Universidad Católica de la Sma, Concepción, emchavez@bond.edu.au; echavez@ucsc.cl

Kieran Conboy Business Information Systems Group, J.E. Cairnes School of Business & Economics, NUI Galway, Ireland Department of Accountancy & Finance, National University of Ireland, Galway, Ireland, kieran.conboy@nuigalway.ie

J. D'Ambra The University of New South Wales, Sydney, NSW, Australia, j.dambra@unsw.edu.au

Jim Davies Oxford University Computing Laboratory, Oxford, UK, jim.davies@comlab.ox.ac.uk

F.J. Domínguez-Mayo University of Seville, Seville, Spain, fjdominguez@us.es

Xiaofeng Du Computer Science Department, University of Durham, Durham, DH1 3LE, England, UK, xiaofeng.du@durham.ac.uk

M.J. Escalona University of Seville, Seville, Spain, mjescalona@us.es

Gavin Finnie School of Information Technology, Bond University, Robina, Gold Coast, QLD, Australia, gfinnie@bond.edu.au

P. Fitsilis Technological Education Institute of Larissa, 41110 Larissa, Greece, fitsilis@teilar.gr

Odd Fredriksson Department of Information Systems, Karlstad University, Karlstad, Sweden, odd.fredriksson@kau.se

Doris Gälle Institute for Applied Informatics, Research Group Application Engineering, University of Klagenfurt, Klagenfurt, Austria, doris@ifit.uni-klu.ac.at

Xiao-hong Gan Information College, Jiangxi University of Finance and Economics, Nanchang 330013, China, xhg1596@163.com

Tejasvi Gaur Department of Computer Science and Computer Engineering, La Trobe University, Melbourne, VIC 3086, Australia, tgaur@students.latrobe.edu.au

Abel Gómez Department of Information Systems and Computation, Polytechnic University of Valencia, Valencia, Spain, agomez@dsic.upv.es

Andrina Granić Faculty of Science, University of Split, Split, Croatia, andrina.granic@pmfst.hr

J.J. Gutiérrez University of Seville, Seville, Spain

Qilong Han College of Computer Science & Technology, Harbin Engineering University, Harbin, China, hanqilong@hrbeu.edu.cn

Lise Tordrup Heeager Aalborg University, Aalborg, Denmark, liseh@cs.aau.dk

Changbing Jiang College of Information Management, Zhejiang Gongshang University, Hangzhou, China, johncabin@mail.zjgsu.edu.cn

Darius Jurkevicius Department of Information Systems, Faculty of Fundamental Sciences, Vilnius Gediminas Technical University, Vilnius, Lithuania, d.jurkevicius@isl.vgtu.lt

Jari Jussila Department of Industrial Management and Engineering, Tampere University of Technology, Pori, Finland

A. Kameas Hellenic Open University, 23 Sahtouri street, 26222 Patras, Greece, kameas@eap.gr

Kestutis Kapocius Department of Information Systems, Kaunas University of Technology, Kaunas, Lithuania, kestutis.kapocius@ktu.lt

Peter Killisperger Competence Center Information Systems, University of Applied Sciences, München, Germany; Advanced Computing Research Centre, University of South Australia, Adelaide, SA, Australia, peter.killisperger@siemens.com

Marite Kirikova Department of Systems Theory and Design, Riga Technical University, Riga, Latvia, marite.kirikova@cs.rtu.lv

Christian Kop Institute for Applied Informatics, Research Group Application Engineering, University of Klagenfurt, Klagenfurt, Austria, chris@ifit.uni-klu.ac.at

Padmanabhan Krishnan School of Information Technology, Bond University, Robina, Gold Coast, QLD, Australia, pkrishna@bond.edu.au

Michael Lang Business Information Systems Group, J.E. Cairnes School of Business & Economics, NUI Galway, Galway, Ireland Department of Accountancy & Finance, National University of Ireland, Galway, Ireland, michael.lang@nuigalway.ie

Even Åby Larsen Department of Information Systems, University of Agder, Serviceboks 422, 4604 Kristiansand, Norway, even.larsen@uia.no

Patricio Letelier Department of Information Systems and Computation, Polytechnic University of Valencia, Valencia, Spain, letelier@dsic.upv.es

Yu Li Institute of Applied Informatics and Formal Description Methods (AIFB), Karlsruhe Institute of Technology (KIT), Universität Karlsruhe (TH), 76128 Karlsruhe, Germany, yu.li@aifb.uni-karlsruhe.de

L.Z. Li Research Center of Cluster and Enterprise Development, Business School, Jiangxi University of Finance and Economic, Nanchang, China

L.X. Li Research Center of Cluster and Enterprise Development, Business School, Jiangxi University of Finance and Economic, Nanchang, China, llx@jxufe.edu.cn

Kirsi Liikamaa Turku School of Economics, Pori Unit, Turku, Finland

Cheng-yu Liu School of Management Science and Engineering, Shanxi University of Finance and Economics, Shanxi, China

Jian Liu School of Information Management, Jiangxi University of Finance & Economics, Nanchang 330013, China, liujian3816@263.net

Lennart Ljung Department of Project Management, Karlstad University, Karlstad, Sweden, lennart.ljung@kau.se

Graham Low School of Information Systems, Technology and Management, University of New South Wales, Sydney, NSW, Australia, g.low@unsw.edu.au

Yun Lu Subei People's Hospital, College of Clinical Medicine, Yangzhou University, Jiangsu 225001, China, yzbzq@163.com

Chunlin Luo School of Information Management, Jiangxi University of Finance & Economics, Nanchang 330013, China, chunlinluo@126.com

Janis Makna Institute of Applied Computer Systems, Riga Technical University, Riga, Latvia, promis@apollo.lv

Nikola Marangunić Faculty of Science, University of Split, Split, Croatia, nikola.marangunic@pmfst.hr

E. Martínez-Force Consejería de Cultura, Junta de Andalucía, Seville, Spain

M. Mejías University of Seville, Seville, Spain, risoto@us.es

Jing-yi Miao School of Management Science and Engineering, Shanxi University of Finance and Economics, Shanxi, China; School of Management, Tianjin University, Tianjin, country, jingyimiao@126.com

Gytenis Mikulenas Department of Information Systems, Kaunas University of Technology, Kaunas, Lithuania, gytenis.mikulenas@ktu.lt

Ivica Mitrović Arts Academy, University of Split, Split, Croatia, ivica.mitrovic@umas.hr

A. Molina Consejería de Cultura, Junta de Andalucía, Seville, Spain

Malcolm Munro Computer Science Department, University of Durham, Durham, NC, USA, malcolm.munro@durham.ac.uk

Elena Navarro Department of Computing Systems, University of Castilla-La Mancha, Spain, elena.navarro@uclm.es

Anders G. Nilsson Department of Information Systems, Karlstad University, Karlstad, Sweden, anders.nilsson@kau.se

Jenny Nilsson Department of Information Systems, Karlstad University, Karlstad, Sweden, jenny.nilsson@msb.se

Thérèse H. Nilsson Universal Child Welfare Foundation (UCWF), Karlstad, Sweden, ucwfoundation@hotmail.com

Ovidiu Noran School of ICT, Griffith University, Nathan, QLD, Australia, o.noran@griffith.edu.au

Andreas Oberweis Institute of Applied Informatics and Formal Description Methods (AIFB), Karlsruhe Institute of Technology (KIT), Universität Karlsruhe (TH), 76128 Karlsruhe, Germany, oberweis@aifb.uni-karlsruhe.de

Tero Päivärinta Department of Information Systems, University of Agder, Serviceboks 422, 4604 Kristiansand, Norway, tero.paivarinta@uia.no

Haiwei Pan College of Computer Science & Technology, Harbin Engineering University, Harbin, China, panhaiwei@hrbeu.edu.cn

Malgorzata Pankowska Information Systems Department, University of Economics, Katowice, Poland, malgorzata.pankowska@ae.katowice.pl

Eric Pardede Department of Computer Science and Computer Engineering, La Trobe University, Melbourne, VIC 3086, Australia, e.pardede@latrobe.edu.au

M. Pérez-Pérez University of Seville, Seville, Spain

Georg Peters Department of Computer Science and Mathematics, University of Applied Sciences, München, Germany

John Sören Pettersson Department of Information Systems, Karlstad University, Karlstad, Sweden, john_soren.pettersson@kau.se

Minna Pikkarainen VTT, Technical Research Centre of Finland, VTT, Finland, minna.pikkarainen@vtt.fi

Maritta Pirhonen Department of Computer Science and Information Systems, University of Jyväskylä, Jyväskylä, Finland

Jaroslav Pokorný Department of Software Engineering, Faculty of Mathematics and Physics, Charles University in Prague, Prague, Czech Republic, jaroslav.pokorny@mff.cuni.cz

Laura Ponisio University of Twente, Twente, The Netherlands, ml@ponisio.com

Isidro Ramos Department of Information Systems and Computation, Polytechnic University of Valencia, Valencia, Spain, iramos@dsic.upv.es

Karel Richta Department of Software Engineering, Faculty of Mathematics and Physics, Charles University in Prague, Prague, Czech Republic, karel.richta@mff.cuni.cz

T. Roach The University of New South Wales, Sydney, NSW, Australia, t.roach@student.unsw.edu.au

Peteris Rudzajs Department of Systems Theory and Design, Riga Technical University, Riga, Latvia, peteris.rudzajs@rtu.lv

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

Kari Smolander Department of Information Technology, Lappeenranta University of Technology, P.O. Box 20, 53851 Lappeenranta, Finland, kari.smolander@lut.fi

William Wei Song Computer Science Department, University of Durham, Durham, NC, USA, w.w.song@durham.ac.uk

Thomas Stückl System and Software Processes, Siemens Corporate Technology, München, Germany

Markus Stumptner Advanced Computing Research Centre, University of South Australia, Adelaide, SA, Australia

Zhen-hua Sun School of Management Science and Engineering, Shanxi University of Finance and Economics, Shanxi, China

Gitte Tjørnehøj Aalborg University, Aalborg, Denmark, gtj@cs.aau.dk

Xiaoqing Tong Hangzhou University of Commerce, Zhejiang Gongshang University, Hangzhou, China, tongxiaoqing321@126.com

Tero Vartiainen Turku School of Economics, Pori Unit, Turku, Finland, tero.vartiainen@oulu.fi

Olegas Vasilecas Department of Information Systems, Faculty of Fundamental Sciences, Vilnius Gediminas Technical University, Vilnius, Lithuania, olegas@fm.vgtu.lt

Doug Vogel Department of Information Systems, City University of Hong Kong, Hong Kong, China, isdoug@cityu.edu.hk

Jürgen Vöhringer Institute for Applied Informatics, Research Group Application Engineering, University of Klagenfurt, Klagenfurt, Austria, juergen@ifit.uni-klu.ac.at

Peter Vrugink Logica, Amsterdam, The Netherlands, peter.vrugink@logica.com

Gui-jun Wang Information Engineering College, Yangzhou University, Jiangsu, 225009, China

Yan Wang Subei People's Hospital, College of Clinical Medicine, Yangzhou University, Jiangsu, 225001, China

Xiaofeng Wang Lero, The Irish Software Engineering Research Centre, Limerick, Ireland, xiaofeng.wang@ul.ie

Xin-chang Wang Information School of Jiangxi University of Finance & Economics, Mathematics and Application Mathematics Department of Jinggangshan University, Jiangxi, China; Mathematics and Application Mathematics Department, Jinggangshan University, Jiangxi, China, wangxinchang11@163.com

Xin-ying Xiao Foreign Languages school, Jinggangshan University, Jiangxi, China, xxinying@163.com

Xiaoqin Xie College of Computer Science & Technology, Harbin Engineering University, Harbin, China, xiexiaoqin@hrbeu.edu.cn

Shenghua Xu School of Information Management, Jiangxi University of Finance & Economics, Nanchang, China, shenghua.xu@jxufe.edu.cn

Zhiqiang Zhang College of Computer Science & Technology, Harbin Engineering University, Harbin, China, zqzhang@hrbeu.edu.cn

Yu-qin Zhao Information Engineering College, Yangzhou University, Jiangsu 225009, China, lanyuzyq@163.com

Part I

**Enterprise Systems – A Challenge for
Future Research and Improved Practice**

CbSSDF and OWL-S: A Scenario-Based Solution Analysis and Comparison

Xiaofeng Du, William Wei Song, and Malcom Munro

Abstract To tackle the semantic issues of web services, we proposed a comprehensive semantic service description framework – CbSSDF – and a two-step service discovery mechanism based on CbSSDF to help service users to easily locate their required services. In this chapter, we evaluate the framework by comparing it with OWL-S to examine how the proposed framework can improve the efficiency and effectiveness of service discovery and composition. The evaluation is done through analysing the different solutions proposed based on these two frameworks for achieving a series of tasks in a scenario.

Keywords Semantic web services · Concept graph · Service description · OWL-S · Semantic match

1 Introduction

In the last decade, enormous research effort of web services has been spent on service description, discovery [10, 11], and composition [1, 4]. In order to effectively and efficiently perform web service discovery and composition, a comprehensive service description framework is essential. There are several semantic service description frameworks that have been proposed to provide richer service descriptions, such as OWL-S [8], WSDL-S [2], and WSMF [7], to address the semantic issue of web services. The main idea behind the existing work is to build a semantic layer either on the top of WSDL or to be integrated into WSDL to semantically describe the capabilities of web services. By having the semantics, a software agent or other services can reason about what a web service's capabilities are and how to interact with it. However, as we mentioned in [5], there are still some problems

X. Du (✉)

Computer Science Department, University of Durham, Durham, DH1 3LE, England, UK
e-mail: xiaofeng.du@durham.ac.uk

that remain in the current semantic service description and search, such as insufficient usage context information, precisely specified requirements needed to locate services, and insufficient information about inter-service relationships.

To tackle the problems, in our previous work [5], we proposed a context-based semantic service description framework (CbSSDF) and a two-step service discovery mechanism to improve the flexibility of service discovery and the correctness of generated composite services. In our recent work [6, 12], we improved the CbSSDF to address more sufficient context information in service descriptions. The context information addressed in the framework is the information that can help to understand the usage of a service and the relationship between the service and other services, so-called service usage context (SUC). By considering the SUC information of services, service discovery can be much more flexible and the service composition process can be simplified.

The main purpose of the chapter is to evaluate the framework by comparing it with OWL-S to examine how the proposed framework can improve the efficiency and effectiveness of service discovery and composition. The evaluation is done through analysing the different solutions proposed based on these two frameworks for achieving a series of tasks in a scenario. The reason we choose OWL-S is that it is the most well known and mature semantic web service description framework and also has been submitted to the World Wide Web Consortium (W3C) for assessment to be a standard.

2 Summary of the Context-Based Semantic Service Description Framework

As discussed previously in [5, 6, 12], to fully describe a service, we must address how the service is related to other services and entities in a business domain and under which context the service should be used. The potential relationship between a service and the other services and entities in a business domain is called service usage context (SUC), which we consider as an important aspect of service description. At a conceptual level, SUC is the conceptual relationships between a service concept and other concepts in a business domain, including other service concepts. At an instance level, SUC is the potential interactions between an instance service and other instance services at runtime. The concept of SUC forms the core of CbSSDF.

There are two layers of components in CbSSDF. The first layer is called service conceptual graphs (S-CGs) that represents the conceptual level SUC. The key point to have S-CGs in a service description is that they bridge the gap between the technical details of services and the conceptual explanations of service users' needs. Normally the users (except the domain experts) are much more familiar to conceptual descriptions than stipulated technical information, such as the information addressed in WSDL. Each S-CG in CbSSDF captures a scenario that the described service can participate in so that it can be matched with service users'

usage scenarios to locate most suitable services. The conceptual graph formalism [13] is used to represent S-CG because (1) conceptual graph provides both flexible visual diagrams for service description and rigour logic expressions and (2) there are well-developed graph matching algorithms for deriving various relations between concepts and relations, such as conceptual graph projection.

The second layer is called semantic service description model (SSDM) that provides enriched semantic service description. SSDM addresses four types of semantics [3] associated with a service, which are data semantics, functional semantics, non-functional semantics, and execution semantics. The data semantics in SSDM is addressed through semantically annotated inputs and outputs. The functional semantics is captured by a service ontology and a set of pre-conditions and effects. The non-functional semantics is addressed through a set of service metadata. The execution semantics is addressed by a description of the internal structure of a service, which also indicates whether a service is a constituent of another service or a service contains a sequence of services.

Another important feature of SSDM is that it embeds the instance level SUC into service description, which can considerably improve the efficiency of service discovery and composition. The key component that addresses the instance level SUC is a set of common usage patterns (CUPs) [6]. Each CUP is a structure that describes how an instance service can be composed with other instance services in terms of semantic compatibility and data compatibility in achieving a task. The whole set of CUPs collectively represents how the instance service can interact/compose with other instance services in its business domain. Therefore, when an instance service is located, the service composition system can easily know which services are compatible with the located service rather than assuming that the services in the entire service repository are compatible and checking them one by one.

In CbSSDF, Defeasible Logic [9] is adopted as the rule language to describe pre-conditions and effect of services and other service composition rules [6]. Defeasible Logic is an implementation of non-monotonic logic, which provides the agile reasoning mechanism for highly dynamic environment and frequently changing conditions, such as the condition reasoning during service composition. The rules in the CbSSDF are divided into two categories: the general rules and the domain-specific rules. The general rules are used to govern and validate the service composition process. The domain-specific rules are used to describe the pre-conditions and effects of services and the business rules and policies in a specific business domain.

We also proposed an improved service discovery mechanism based on CbSSDF: a two-step service discovery mechanism [5]. The S-CGs in CbSSDF support the service search engine to locate services by concepts and conceptual relations, so in the first step what a service user needs to do is to describe his requirements or usage scenarios in natural language without worrying about any technical details. The search engine will first convert the natural language query into a CG and match with the S-CGs in the service repository to locate the query's relevant services. The located services may or may not be the exact required services. However, it guarantees that these services are relevant to the user's query. The second step is to refine

the result from the first step using the technical specifications given by the service user, generate composite services, and rank the result according to the similarity degree to the specifications. The significant difference between the two-step service discovery mechanism and the traditional service discovery methods is that after the first step, it is easier for the service users to propose detailed technical service specifications, and the service composition process can be much more efficient due to the instance level SUC information in SSDM. The two-step service discovery mechanism demonstrates how the features provided by CbSSDF can facilitate the service discovery and composition.

3 The Evaluation Scenario

Semantic web service description, discovery, and composition are brand new research topics and still in their premature stages. There is no commercially released software or tools that comprehensively tackle these areas. There are only a few research-based prototypes and APIs, such as the OWL-S API. In other words, there is no existing system that can be directly compared with our solution. Therefore, we set up a scenario with three tasks and give a virtual analysis to see how our solution can achieve the tasks differently from other semantic web service solutions and examine if our solution can improve the situation. The existing semantic web service description framework we choose to compare with CbSSDF is OWL-S.

The scenario for comparing the differences between the CbSSDF-based solution and the OWL-S-based solution is to perform a compound arithmetic calculation using mathematical web services. Normally, an arithmetic calculation is carried out by a combination of symbols, mathematical operators, and certain rules. We use a web service to represent a mathematic operator, where the service inputs are the operator's operands and the output of the service is the result of the calculation. A compound arithmetic expression can be represented as a composite service, where each operator in the expression is considered as a participant web service of the composite service. The calculation result is produced through executing the participant services in the composite service following certain mathematical rules. Although using web services to perform arithmetic calculation is not a complex scenario, it tackles all the aspects of service discovery, composition, and invocation. It involves service query (either in natural language or formal mathematical expression) processing, service discovery based on semantics and technical specifications, data-type compatibility checking during service composition, service planning, and rule-regulated service composition and invocation. Therefore, it requires a service description framework that provides sufficient information in order to achieve the above tasks with minimum human intervention. By going through the tasks from the scenario, we will assess which solution can make the tasks easier to achieve with the least human intervention.

Suppose we have a student who wants to find a web service to calculate the volume of a cone. He knows how this can be done, but he wants a web service to do it for him. He proposes a natural language query as follows:

“Cone volume calculation service: multiply a cone’s base circle area by its height and divide by 3”.

This query states which kind of service he is looking for and how the service should work. Now let us analyse what the possible situations of the returned results are:

- One or many existing atomic services from the service repository are located for the requirement.
- One or many existing composite services from the service repository are located for the requirement.
- There are no existing services that can satisfy the requirements, but a composite service is constructed dynamically for the requirement.
- No (satisfiable) result returned, i.e. neither existing services nor dynamically constructed composite services can fully satisfy the requirement.

The last situation will not be discussed and the rest three situations will be set as three tasks to see how each of the two solutions can achieve them. Suppose we have the query interfaces for both CbSSDF- and OWL-S-based solutions and a service repository containing mathematical web services that represent different arithmetic operators and calculations. Some of them are atomic services, such as addition service, multiplication service, π value service, and square root service. Some of them are composite services, such as circle area calculation service and cylinder volume calculation service. Both the atomic services and the composite services can be composed to construct more complicated composite services. Two examples of the CbSSDF-based service description are shown in Table 1.

Most of the information in Table 1 is obvious. The “CUP input services” means that these services can provide input data for the described service. The “CUP output services” means that these services can consume the output data from the described service. The mathematical services do not have strong data semantics restriction, thus in the above case any services in the service repository that can provide or consume *double* data type are in the “CUP input services” list or the “CUP output services” list. An atomic service does not have internal structure, hence the addition service’s “internal structure” is *null*. OWL-S-based service description examples can be found on MindSwap website¹.

4 Tasks Analysis and Results

In the following sections, we will set three tasks and give the solution analysis result. The three tasks are (1) locating an existing atomic service; (2) locating an existing composite service; (3) dynamically constructing composite service.

¹ MindSwap OWL-S example: <http://www.mindswap.org/2004/owl-s/services.shtml>

Table 1 Two examples of the CbSSDF-based service description

	Atomic service	Composite service
Name	Addition service	Circle area service
Type	Arithmetic	Area
Input data type	In1: double In2: double	In1: double In2: double
Input semantics	In1: addend In2: summand	In1: radius In2: pi
Output data type	Out1: double	Output1: double
Output semantics	Out1: summation	Out1: circle area
Pre-condition	isDouble(in1) $\wedge$ isDouble(in2)	isDouble(in1) $\wedge$ isDouble(in2)
Effect	isDouble(out1)	isDouble(out1)
CUP input services	In1: subtraction, division, ... In2: subtraction, division, ...	In1: subtraction, division, ... In2: subtraction, division, ...
CUP output services	Subtraction service, division service, ...	Subtraction service, division service, ...
Internal structure	Null	Multiplication, square, and PI
Metadata	QoS, natural language description, service provider detail, ...	QoS, natural language description, service provider detail, ...
Resource	Not provided	Not provided
S-CGs	[Arithmetic: addition service] $\leftarrow$ (REQ) $\leftarrow$ [Area: trapezium area service] $\rightarrow$ (REQ) $\rightarrow$ [Arithmetic: multiplication service]	[Area: circle area service] $\leftarrow$ (REQ) $\leftarrow$ [Volume: cylinder volume service]

4.1 Task 1: Locating an Existing Atomic Service

In the first task, we assume that there is at least one atomic service in the service repository that can satisfy the requirements, i.e. perform the calculation of the volume of a cone. This task requires that a service description framework has the capability to support query interpretation and specification matchmaking.

4.1.1 Solution Comparison

The comparison result of these two different solutions for solving task 1 is listed in Table 2.

4.1.2 Summary

From the comparison result shown in Table 2, we see that for locating a single atomic service, there is no significant difference between these two solutions. They both require the service user to propose a query followed by a detailed technical specification of the required service. However, in the CbSSDF-based solution, the matchmaking can be performed based on imprecise information, such as the natural language query. The service user can provide technical specification later based on

Table 2 The comparison of CbSSDF- and OWL-S-based solutions for task 1

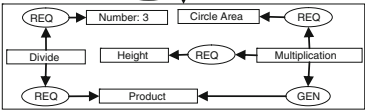
CbSSDF-based solution	OWL-S-based solution
1. Query interpretation A given query Q is converted into a CG: $Q \Rightarrow CG$ 2. Match making Step one: From CG matching a set of relevant services $S_r = \{s_1, s_2, \dots, s_n\}$ can be obtained. Then the services in S_r are ranked according to their S-CGs' similarity to the query CG. Step two: Based on the further technical specification provided by the service user, S_r is refined, ranked according to similarity, and returned to the service user. The specification matching is performed based on the attributes addressed in SSDM, such as IOPE, service concept, metadata, internal structure, and CUPs.	1. Query interpretation A given query Q is converted into a set of concepts: $Q \Rightarrow C = \{c_1, c_2, \dots, c_n\}$ 2. Match making Not applicable. Matchmaking cannot be performed based on natural language query. Therefore, the technical specification is required at the same time when the query is proposed. The matchmaking is performed based on IOPE and metadata. A set of result service is returned to the service user and they are ranked according to the similarity to the specification.

the interim results. The OWL-S-based solution requires the service user to give technical specification of required services at the very beginning of the search. It could be a difficult task for the service user to give detailed technical specification of a required service, especially when the service user is not a domain expert in the required service area.

4.2 Task 2: Locating an Existing Composite Service

In the second task, we assume that in the service repository there is at least one composite service that can perform the calculation of the volume of a cone. This task requires that a service description framework has the capabilities to support query interpretation, specification matchmaking, and internal structure and sub-services matching with composite services if applicable.

Table 3 The comparison of CbSSDF- and OWL-S based solutions for task 2

CbSSDF-based solution	OWL-S-based solution
1. Query interpretation: A given query Q is converted into a CG: $Q \Rightarrow CG$  2. Matchmaking Step one –CG matching: From CG matching a set of relevant services $S_r = \{s_1, s_2, \dots, s_n\}$ can be obtained. Then the services in S_r are ranked according to their S-CGs' similarity to the query CG. Step two – specification matching: Based on the further technical specification provided by service users, S_r is refined, ranked, and returned to the service user. However, in this step, if the service user is familiar with the required services and able to provide the internal sub-services' detail, the result can be more accurate. For example, if service s is a composite service and consists of s_i and s_j, then s_i and s_j's details can also be used to locate s.	1. Query interpretation: A given query Q is converted into a set of concepts: $Q \Rightarrow C = \{c_1, c_2, \dots, c_n\}$ 2. Matchmaking Not applicable. The internal details of services are hidden from service users in OWL-S-based solution, thus for a service user, the atomic service and the composite service are not distinguished. For this reason, the matchmaking process in this task is exactly the same as the one described in task 1.

4.2.1 Solution Comparison

The comparison result of these two different solutions for solving task 2 is listed in Table 3.

4.2.2 Summary

The comparison result in Table 3 shows the difference between these two solutions in dealing with composite service discovery. In the CbSSDF-based solution, the rich service description enables service users to be able to use extra information, such as the internal structure composite services, to more precisely locate required services. In reality, it is not necessary for a user to know whether the required service is an atomic service or a composite service. However, if the user does know, the extra information can be used to obtain better search result. In the OWL-S-based solution, composite service and atomic service are not distinguished from the service user's perspective. The advantage of this is that it makes the service discovery and service description simpler. From the CbSSDF point of view, it tries to use all the available information to assist service discovery. The disadvantage of this is that it increases the complexity of service description and discovery. However, the complex service description can be compensated by more accurate service discovery result.

4.3 Task 3: Dynamically Constructing Composite Service

In the third task, we assume that there is no existing service that can perform the calculation of the volume of a cone. Therefore, a composite service needs to be dynamically constructed by using the existing services in the service repository.

This task requires that a service description framework has the capabilities to support not only query interpretation and specification matchmaking, but also service planning during the service composition process.

4.3.1 Solution Comparison

The comparison result of these two different solutions for solving task 3 is listed in Table 4.

4.3.2 Summary

In the situation of dynamic composite service construction, the CbSSDF-based solution fully reveals its advantages. Compared to OWL-S-based solution, the information provided by CbSSDF can much improve the performance of service composition and the accuracy of the result. The inter-relationship between services addressed in CUPs can narrow down the number of candidate services in each step of planning. As each CUP can be considered as a segment of a plan, the actual steps of a plan to reach the desired goal is also reduced. The general rules and domain-specific rules can be used to not only describe the pre-conditions and effects of services, but also inspect the correctness of generated composite services.

In OWL-S-based solution, each service has been treated completely individually. The potential inter-relationship between services has been ignored. One of the consequences of this is slow performance because during the planning phase, each task's possible candidate services are the entire collection of services in the service repository rather than the services that are relevant and compatible to the service in the previous task.

4.4 Discussion of the Scenario-Based Comparison

In this section, we used an arithmetic calculation scenario to compare the CbSSDF-based solution with the OWL-S-based solution. Three tasks have been proposed to examine how each solution deals with the following situations:

- *Locating an atomic service:* In this situation, the service description framework needs to assist service search engine to locate the existing atomic services that can fulfil the service user's requirements.
- *Locating a composite service:* In this situation, the service description framework needs to assist service search engine to locate the existing composite services that can fulfil the service user's requirements.

Table 4 The comparison of CbSSDF- and OWL-S-based solutions for task 3

CbSSDF-based solution	OWL-S-based solution
1. Query interpretation A given query Q is converted into a CG: $Q \Rightarrow CG$	1. Query interpretation A given query Q is converted into a set of concepts: $Q \Rightarrow C = \{c_1, c_2, \dots, c_n\}$
2. Matchmaking Step one –CG matching: From CG matching a set of relevant services $S_r = \{s_1, s_2, \dots, s_n\}$ can be obtained. The set of services are ranked according to their S-CGs' similarity to the query CG. The CG matcher will also join single S-CGs together into larger S-CGs in order to achieve maximum match. Step two – specification matching: Based on the further technical specification provided by service users, S_r is refined and ranked. If there is no matched service or matched with very low similarity rate, the system will start service planning to generate composite services.	2. Matchmaking Not applicable.
3. Planning and composition – The planning is based on reduced service range S_r, i.e. the relevant services are considered first in the planning. – When a service is located, its CUPs tell the planner where to go next. Only the services in its CUPs are compatible, therefore, there is no need to go through all the services in the repository.	Based on the further technical specification provided by service users, the OWL-S-based solution will try to find a set of best-matched services. If there is no matched service or matched with very low similarity rate, service composition will be attempted. 3. Planning and composition – The planning is based on assuming all the services in the entire service repository are candidate services. – The planner in OWL-S-based solution needs to go through the entire service repository every time when locating a service for a task.
4. Rule evaluation – The pre-conditions and effects of each service are evaluated during service planning process. – The general rules and domain-specific rules are evaluated to filter out invalid composite services.	4. Rule evaluation – The pre-conditions and effects of each service are evaluated during service planning process.

- *Dynamically constructing a composite service:* In this situation, there is no existing service that can fulfil the service user's requirements. Therefore, the service description framework needs to assist service search engine to dynamically construct composite services to fulfil the service user's requirements.

By analysing the two different approaches through the three tasks, we have found the pros and cons of these two solutions summarised as follows:

1. In dealing with the atomic service discovery, these two solutions have no considerable differences. However, in the CbSSDF solution, the two-step service discovery mechanism gives service users the flexibility to search services by imprecise information first, such as natural language, and then gives the precisely specified technical information later to refine the result. In the OWL-S solution, the service users have to provide precisely specified technical information at the very beginning of each search, which may be hard for the users who are not familiar with the technical details of the required services.
2. In OWL-S solution, the composite service and the atomic service are not distinguished in service user's perspective. Therefore, for a service user, there is no difference in the way he searches for an atomic service or a composite service. The advantage of this is that it makes the searching process simpler and the user need not be aware of the difference between composite services and atomic services.
3. One of the principles of the CbSSDF is to use all the possible information to assist service discovery. Therefore, if the service users do know the internal detail of composite services they are looking for, they can provide relevant information, which can make the discovery result more accurate. However, the disadvantage is that it increases the complexity of service description.
4. When the situation is getting complicated, i.e. when dynamically composite services construction is required, the advantages of the CbSSDF-based solution emerge. First of all, the two-step service discovery mechanism can filter out the irrelevant services by CG matching so that the candidate services for service composition are reduced. Second, the CUPs can further reduce the number of candidate services in each step of a service planning process. They can also reduce the number of steps that a planner needs to reach the goal state. Third, the non-monotonic rules in the CbSSDF can help to identify invalid composite services, which ensure the result services are correct and executable.

A significant defect of OWL-S solution is that it does not consider the inter-relationship of services. The consequence is that no matter in which stage of a service planning process, the planner has to search through the entire service repository for candidate services.

5 Conclusion

To tackle the semantic issue of web services, we proposed a comprehensive semantic service description framework – CbSSDF – and a two-step service discovery mechanism based on CbSSDF in our previous work to help service users to easily locate their required services. The main purpose of this chapter is to compare the CbSSDF-based solution with OWL-S through a series of tasks in a designed

scenario in order to analyse the advantages of the CbSSDF in service discovery and composition. The analysis result shows that the CbSSDF-based solution can improve the users' service searching experience by the two-step service discovery mechanism and increase the efficiency and effectiveness of service discovery and composition by embedding SUC and richer semantics in SSDM. The major defect of the CbSSDF is the complexity because it addresses richer semantics and the SUC information. However, the more accurate service discovery results compensate the complexity defect.

Performance evaluation has not been addressed in this chapter. However, in our previous work [6] we have proposed some evaluation results on service discovery result accuracy, system performance, and system scalability.

References

1. Agarwal, S., Handschuh, S., and Staab, S. (2005) Annotation, Composition and Invocation of Semantic Web Services, *Journal of Web Semantics* 2(1).
2. Akkiraju, R., Farrell, J., Miller, J., Nagarajan, M., Schmidt, M., Sheth, A., and Verma, K. (2005) Web Service Semantics – WSDL-S, A Joint UGA-IBM Technical Note, version 1.2, April 18, 2005, <http://lsdis.cs.uga.edu/projects/METEOR-S/WSDL-S>.
3. Cardoso, J., and Sheth, A. (Eds.) (2006) *Semantic Web Services, Processes and Applications, Semantic Web and Beyond Computing for Human Experience Vol. 3*. Springer.
4. Du, X., Song, W., and Munro, M. (2006) Using Common Process Patterns for Semantic Web Services Composition, to appear in *Proceedings of 15th International Conference on Information System Development (ISD2006)*, Budapest, Hungary, August 31–September 2, 2006.
5. Du, X., Song, W., and Munro, M. (2007) Semantic Service Description Framework for Addressing Imprecise Service Requirements, in *Proceedings of 16th International Conference on Information System Development (ISD2007)*, Galway, Ireland, August 29–31, 2007.
6. Du, X., Song, W., and Munro, M. (2008) A Method for Transforming Existing Web Service Descriptions into an Enhanced Semantic Web Service Framework, in *Proc. of 17th International Conference on Information System Development (ISD2008)*, Paphos, Cyprus, August 25–27, 2008.
7. Fensel, D., and Bussler, C. (2002) The Web Service Modeling Framework WSMF, *Electronic Commerce Research and Applications* 1(2): 113–137.
8. Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., Narayanan, S., Paolucci, M., Parsia, B., Payne, T., Sirin, E., Srinivasan, N., and Sycara, K. (2004) OWL-S: Semantic Mark-up for Web Services, <http://www.daml.org/services/owl-s/1.0/owl-s.html>.
9. Nute, D. (1994) Defeasible logic, in *Handbook of Logic in Artificial Intelligence and Logic Programming* (vol. 3): *Non-Monotonic Reasoning and Uncertain Reasoning*, Oxford University Press.
10. Paolucci, M., Sycara, K., and Kawamuwa, T. (2003) Delivering Semantic Web Services, in *Proceedings of WWW2003*, Budapest, Hungary, May 20–24, 2003, pp. 829–836.
11. Song, W., and Li, X. (2005) A Conceptual Modeling Approach to Virtual Organizations in the Grid, to appear in *Proceedings of GCC2005* (eds. Zhuge and Fox), Springer LNCS 3795, pp. 382–393.
12. Song, W., Du, X., and Munro, M. (2009) A Concept Graph Approach to Semantic Similarity Computation Method for e-Service Discovery, to appear in *International Journal of Knowledge Engineering and Data Mining*, InderScience publishers.
13. Sowa, J. F. (1984) *Conceptual Structures: Information Processing in Mind and Machine*, Addison-Wesley, Canada.

Enterprise Systems in a Service Science Context

Anders G. Nilsson

Abstract By enterprise systems we here refer to large integrated standard application packages that fully cover the provision of information required in a company. They are made up of extensive administrative solutions for management accounting, human resource management, production, logistics and sales control. Most of the enterprise systems on the market have traditionally been designed with a focus on manufacturing companies, but during the past years the supply of various enterprise systems for service-oriented business organizations has gradually increased. This fact raises the issue to study enterprise systems from a service management perspective. Service science is an emerging discipline that studies value creation through services from technical, behavioural and social perspectives. Within service science it is therefore possible to use and apply a wide spectrum of engineering tools for development of business services in organizations. In this sense, enterprise systems represent an efficient tool for service innovations. The research interest in this chapter is focussed on how we can study enterprise systems in a service science context.

Keywords Enterprise systems · Service science · IT artefacts · Business reshaping · Life cycles · Method support · Levels of change

1 Enterprise Systems

A current trend since the middle of 1990s is that a growing number of business software packages are classified as enterprise systems [9, 21]. Enterprise systems have been used as advanced tools to increase the business capacity in companies and organizations [2]. By enterprise systems we here refer to large integrated standard application packages that fully cover the provision of information required in a company. Enterprise systems are made up of extensive administrative solutions for

A.G. Nilsson (✉)

Department of Information Systems, Karlstad University, Karlstad, Sweden

e-mail: anders.nilsson@kau.se

management accounting, human resource management, production, logistics and sales control. An important criterion is that the included parts are closely integrated with each other through a central database [8]. From that standpoint we can conclude that enterprise systems are all-embracing IT supports for the whole business in companies and organizations [17].

An advantage of enterprise systems is that the vendor guarantees that different functions in the business software package are connected, with thoroughly tested interfaces. A disadvantage is that the different parts in the vendor's enterprise system are often of varying quality. For this reason it may be wise to combine an enterprise system with one or more niche packages. Because of their extensiveness, enterprise systems usually are called enterprise resource planning (ERP) systems [17, 29].

Nowadays many organizations are facing a complex existence, with mixed system environments (platforms) and multiple IT solutions for the same applications in the business. It is not unusual in large companies to find perhaps five different material resource planning (MRP) packages running parallel – often operating on different platforms – as a result of previous organizational mergers. It is therefore tempting to start afresh, replacing existing IT solutions by a new, fresh enterprise system. Most of the enterprise systems on the market have traditionally been designed with a focus on manufacturing companies, but during the past years the supply of various enterprise systems for service-oriented business organizations has gradually increased [23]. This fact raises the issue to study enterprise systems from a service science perspective.

2 Service Science

The service sector dominates the global economy of today. Services have come to represent more than 75% of the gross domestic product of developed nations [16]. In most countries, services add more economic value than agriculture, raw materials and manufacturing combined. In developed economies, employment is dominated by service jobs and most new job growth comes from services. Jobs range from high-paid professionals and technicians to minimum-wage positions. Most activities by government agencies and non-profit organizations involve services [28].

Service science is an emerging discipline that studies value creation through services from technical, behavioural and social perspectives [12]. This new discipline is the application of services management and engineering sciences to work tasks that one organization beneficially performs for and with their customers. Service science is truly multidisciplinary and builds on knowledge and experience developed from marketing, operations management, sociology, psychology, working life science, computer science and information systems [18]. According to one of the pioneers, Jim Spohrer, the new discipline of service science (from around 2005) could be described as follows [5]:

Service Science is the short term for Services Sciences, Management, and Engineering (SSME). This new discipline is the application of scientific, management and engineering disciplines to tasks that one organization beneficially performs for and with another ('services'). Science is a way to create knowledge. Engineering is a way to apply knowledge

and create new value. Management improves the process of creating and capturing value. Service Science is truly multidisciplinary! (Jim Spohrer, Director of Services Research, IBM Almaden Research Center, San Jose, California)

Service science as a discipline focusses on fundamental science, models, theories and applications to drive innovation, competition and quality of life through services [4]. This definition suggests a focus on substantive outcomes (innovation, competition and quality of life), grounded in rigorous research (science, models, theories and applications). The definition does not preclude any relevant discipline from participating, nor does it prescribe a particular type of research methodology.

Service science has a potential to stimulate a new and fruitful cooperation between scholars within different academic disciplines to develop concepts, models, theories and not least relevant empirical studies on value creation through service. Service science should open up for and invite scholars in areas such as software metrics and software development, service-oriented architecture (SOA), open source frameworks, service simulation, system interaction and integration, service management control and business strategy [27]. The focus should be on how value is co-produced or co-created with customers and thus add value for other stakeholders such as shareholders, employees and society in general. Both strategic and operational issues must be focussed on [19].

Within service science it is therefore possible to use and apply a wide spectrum of engineering tools for development of business services in organizations [7]. In this sense, enterprise systems represent an efficient tool for service innovations. Enterprise systems and service science do have intersections today which will widen due to the fact that more and more IT vendors are trying to take up the SOA (service-oriented architecture) paradigm [11]. We will now look closer to possible connections between enterprise systems and service science in a business environment.

3 Connections Between Enterprise Systems and Service Science

For gaining a deeper understanding of the practical use of IT systems in service-based organization it is essential to highlight the relationships between enterprise systems and service science from a business environment standpoint (see Fig. 1). Enterprise systems and service science depend on each other. Particularly, they both influence each other for various connections.

The relationships can also be regarded from a Venn diagram perspective with joint and separate parts. Enterprise systems are general or universal IT solutions for all kinds of organizations, e.g. for traditional manufacturing companies as well as for professional service-oriented organizations. On the other hand, service science as a discipline can propose many kinds of engineering tools where, e.g., enterprise systems could give good potentials for supporting service management in organizations.

The research interest is here focussed on how we can study enterprise systems in a service science context. We have approached this issue from investigating some

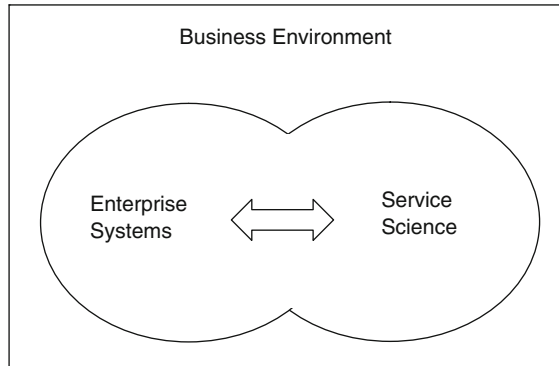


Fig. 1 The relationship between enterprise systems and service science

possible connections [20–22] or significant conditions between enterprise systems and service science related to

- IT artefacts
- Business reshaping
- Life cycles
- Method support
- Levels of change

We will describe each possible connection in a separate section below. These connections have been found when analyzing the literature of enterprise systems and service science as well as using practical experiences from business projects.

4 Enterprise Systems and Service Science – IT Artefacts

Enterprise systems could be regarded as useful IT artefacts to support some kind of business in organizations. By IT artefacts we mean the use of hardware and software solutions to improve the business activities and service processes within and between organizations [24]. The IT artefacts can be of a varied character, for example, we can create IT solutions in organizations by using software metrics, service-oriented architecture (SOA), open source frameworks, service simulation and system integration (see Section 2). We are here focussing on enterprise systems as IT artefacts for developing and changing the situation in concrete business service cases. In the service science literature you will find IT outsourcing service systems [28] and offshore outsourcing [4] as other examples of IT artefacts for service innovations.

Enterprise systems are examples of IT systems for the collection, processing, storage, retrieval, distribution, presentation and use of information in organizations.

An enterprise system is an integrated part of the business operations that it is supposed to serve or in other words an embedded system with the business services in companies [1]. It is not an end in itself, but intentionally arranged for organizing the message exchange or communication between people for supporting their work tasks in service organizations [7]. Enterprise systems could also have a more offensive or aggressive target for enabling or creating new business opportunities in service companies, e.g. Internet Banking and Electronic Commerce. In the new service economy, enterprise systems will play an essential role for promoting a more proactive service management [13].

The ISD subject has a tradition to be multidisciplinary in character trying to study the phenomenon of “information systems” such as enterprise systems from, e.g. technical, economical and pedagogical aspects. Therefore, there is a need to integrate knowledge from different disciplines, such as computer science, business administration and behavioural science [6], when studying the phenomenon of enterprise systems in organizations. Therefore, the disciplines of information systems and service science have much in common as multidisciplinary subjects [27].

A significant condition is that organizations live with enterprise systems in an increasingly changing world. There are a number of trends or driving forces in the business world around us that will have a growing impact on investments in enterprise systems [22], for example:

- The structure of companies is becoming more virtual, horizontal and network-based.
- Enterprise systems are to a greater extent used as inter-organizational or business-to-business (B2B) solutions between service companies.
- Actors are increasingly operating on electronic or digitized markets using the Internet technology and E-business framework for service organizations.

In this light, the fields of information systems and service science will play an increasingly important part in the future. We need to invest in enterprise systems for the professional service organizations of tomorrow. This is a position statement on service science from an information systems perspective where the IT artefact is represented by an enterprise system.

5 Enterprise Systems and Service Science – Business Reshaping

From earlier experience we have noticed that enterprise systems are going through different stages or phases in business reshaping of service operations in companies and organizations [22, 30]:

1. Automation and Efficiency
2. Integration and Cooperation
3. Transformation and Networking

In the *first stage* the focus is on automating certain service operations, to do things right, faster and cheaper with support of enterprise systems. The primary use of enterprise systems has been to increase the efficiency of different functions or activities in organizations, e.g. by automating service jobs that earlier was carried out manually. This approach could lead to “information islands” in service organizations more or less isolated from each other.

In the *second stage* the focus is on cooperation between business processes and service operations inside companies – from these viewpoint, efficient functions or activities are important but not sufficient. Business people often think more in terms of workflows or service processes for achieving expected results. Enterprise systems as an integrated support to service operations become a key issue on the top management agenda. This approach promotes that bridges are being built between “information islands” in service organizations.

In the *third stage* the focus is on transforming service operations in the market place for creating competitive power of the enterprise systems. The value constellation or networks of business actors comprise our focal company, customers, clients, suppliers and partners. Enterprise systems from different business actors are linked to each other and shared databases or e-portals are used. This approach supports inter-organizational solutions and connects “information islands” over company boundaries for service organizations.

An interesting condition is that all three stages, automation, integration and transformation, are interdependent, which means that we must work with them simultaneously. The “field of play” is to go through the three stages of business reshaping over and over again to make an improved use of enterprise systems in service organizations. The value creation of a company is performed by the business services [12, 19] in interaction with the enterprise systems in use. In this sense, enterprise systems must be considered in a service science context.

6 Enterprise Systems and Service Science – Life Cycles

Change work in organizations 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. We will here focus on life cycles for investments in enterprise systems and for development of business services in organizations.

A life cycle can be partitioned in a number of phases or areas. On a crude level a development life cycle 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 therefore demand various bodies of knowledge and professional competence [20].

What pattern lies behind a life cycle philosophy? Development work can be seen as a form of decision-making activity. The Nobel Prize winner Herbert Simon (in

1978) states that all kinds of decision making go through three phases: intelligence (I), design (D) and choice (C) [26]. 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). A general model for change processes (based on IDC) comprises 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 of enterprise systems [20]. Strictly speaking, by information systems development (ISD) we mean the acquisition phase including steps for analysis, design and implementation of IT artefacts.

From an *enterprise systems perspective* we can identify three different types of life cycle models [15]. From the beginning there has been a customized development of an in-house system for a specific company (original development model). Further on an IT vendor has performed a generalized and packaged solution in the shape of a newborn enterprise system for sale on an open market (vendor development model). Thereafter every customer or user organization has to perform an acquisition and implementation of a selected enterprise system among the IT vendors (customer development model). This last life cycle model is what we normally think of when adopting and deploying enterprise systems in service organizations [3].

From a *service science point of view* there are several life cycle models launched in the literature for service innovation and development. In a work system life cycle model for business services we can identify four different phases: initiation, new development, implementation and operation/maintenance of service systems for organizations [1]. In a discipline life cycle model for service science we can identify three comprehensive phases: strategic planning, innovative design and operation/evolution of business services [14]. In a service research life cycle model we can identify four overall phases for service development and innovation: service idea generation, the service strategy and culture gate, service design and service policy deployment and implementation [13].

It is now an interesting task to try to compare and contrast life cycle models from general information systems development (ISD), acquisition and use of enterprise systems in customer organizations as well as present experiences from the service science area (see Table 1). It is here an attempt to sketch a four-phase model for each of the three areas based on the presentation of different life cycle approaches above.

Table 1 Life cycles from an ISD, enterprise systems and service science perspective

ISD life cycle	Enterprise systems	Service science
1. Analysis	1. Selection	1. Planning
2. Design	2. Customization	2. Innovation
3. Construction	3. Adoption	3. Evolution
4. Assessment	4. Management	4. Operation

There is a need for life cycle models for development and use of enterprise systems as well as for service development and innovation of professional organizations. When studying different life cycle models for enterprise systems and business service development (from service science) they seem to have more similarities than discrepancies. This is an interesting observation which facilitates a desirable integration and interaction between enterprise systems and business services. Earlier research and practice from our ISD area could here be of valuable support for elaboration and extension of future life cycle models.

7 Enterprise Systems and Service Science – Method Support

Reliable experience shows that issues concerning the design and use of enterprise systems in organizations need to be addressed systematically [21]. Nevertheless, quite often the investments in enterprise systems are performed following ad hoc strategies. Enterprise systems or ERP systems are implemented into more or less chaotic company environments, where too much happens at once. Business people tend to select enterprise systems by instinct behaviour (using their “heart”) rather than by rational thinking (using their “brain”). Some of the effects of this could be as follows:

- The enterprise systems are underused, or even disrupt the business services of the company.
- An increased IT vendor dependency, which leads to extensive extra work for the service providers in the organization.
- Constant adaptations are made, both in the business services and the enterprise systems.

Earlier research has to some degree been focussing on systematic ways of working or method support for acquiring, implementing and maintaining enterprise systems in organizations. The traditional approach in the ISD discipline is to support with general guidelines and checklists for managing enterprise systems in companies [20]. From services research we have recognized a complementary approach with different supporting methods for service idea generation, service strategy and culture, service design and service policy deployment [12, 13]. From IS research we have recently developed method support for characterizing business services as work practices and communication patterns [7] which will make the use of enterprise systems more transparent in service organizations.

A good working principle is to be able to combine systematics with inspiration in a sensible manner when implementing enterprise systems in our service organizations. We need appropriate “doses” of both methodology and creativity to achieve successful results when designing new business services in an effective interplay with applied enterprise systems. In this regard, enterprise systems have to be understood within a service science discipline.

8 Enterprise Systems and Service Science – Levels of Change

Enterprise systems should be viewed in a wider organizational context. Business performance of service management generally consists of different tasks which can be collected into some appropriate levels [1, 10, 11, 13]. We can recognize three levels of change for work practices in companies, each level with a distinct scope and focus [22]:

- **Business market level**

Focussing on strategies for improving the business relationships between our company and the cooperating actors in the market environment.

- **Service operation level**

Focussing on strategies for making service operations more efficient within our company; the workflow or processes are improved.

- **Enterprise systems level**

Focussing on strategies for how enterprise systems can be more useful resources for running the service operations more professionally and competitively.

In today's business world, information support by enterprise systems has become a more integrated part of service operations and, in many cases, a vital part of the business mission itself. In fact, the enterprise systems could also create new service opportunities for companies to reinforce their competitive edge in the market place. In many cases development of business markets, service operations and enterprise systems are often carried out as separate change measures and as independent projects in organizations.

The business challenge is to have a proper organizational coordination and timing between the three levels of work practices in companies. Strategic congruence and integrated control between organizational levels are essential issues on the top management agenda in companies. Therefore, investments in enterprise systems should be in harmony with the efforts taken on business market and service operations levels in organizations. In other words, enterprise systems should be regarded from a service science perspective.

9 Summing Up

The research interest in this chapter is focussed on how we can study enterprise systems in a service science context. We have identified and described five possible connections between enterprise systems and service science as follows:

- Investments in enterprise systems as *IT artefacts* represent a valuable engineering tool for making service organizations more effective in the business world.
- Investments in enterprise systems could strengthen the stages of *business reshaping* in service organizations (i.e. automation, integration, transformation).

- Investments in enterprise systems follow a *life cycle* model which should be coordinated with the corresponding life cycle model for service development.
- Investments in enterprise systems for service organizations should be guided by a systematic way of working with a solid *method support* for change work.
- Investments in enterprise systems should be in harmony with different *levels of change* in service organizations (i.e. market, operation and systems levels).

These possible connections between enterprise systems and service science are grounded from a series of theoretical and empirical studies based on a scientific design method called consumable research [25]. Most of the enterprise systems on the market have traditionally been designed with a focus on manufacturing companies, but during the past years the supply of various enterprise systems for service-oriented business organizations has gradually increased. Therefore, it is important to study and investigate enterprise systems in a service science context.

In conclusion I would like to give our ISD discipline a challenge for the future. In this respect I will refer to a well-known formula for performing success in business services by applying it to the area of information systems development:

$$\text{Degree of success in ISD} = f(\text{Quality} \times \text{Acceptance} \times \text{Value})$$

The success formula states that to attain a successful result for information systems development (ISD) in organizations, we must have a sufficient quality in the designed IT solutions and a good acceptance among the users or people to give them a motivation for using the enterprise systems as well as that the designed IT solutions should create a business value to the ultimate beneficiaries or customers to the company. A low figure in quality, acceptance or value will lead to an unsuccessful result – hence, the multiplication sign in the success formula. There is a strong interplay between computers, people and work tasks in organizations – a lesson learnt from the history of the ISD discipline! As a final point we would like to give service organizations a real business challenge for the future:

Be open to the new and interesting opportunities that modern enterprise systems offer.
Winners are those who make the best use of the enterprise systems in their business services!

References

1. Alter, S. (2008) Service System Fundamentals: Work System, Value Chain, and Life Cycle, *IBM Systems Journal*, 47(1): 71–85.
2. Arnold, V. (2006) Behavioral Research Opportunities: Understanding the Impact of Enterprise Systems, *International Journal of Accounting Information Systems*, 7(1): 7–17.
3. Benders, J., Batenburg, R., and van der Blonk, H. (2006) Sticking to Standards: Technical and Others Isomorphic Pressures in Deploying ERP-Systems, *Information & Management*, 43(2): 194–203.
4. Bitner, M. J., and Brown, S. W. (2006) The Evolution and Discovery of Services Sciences in Business Schools, *Communications of the ACM*, 49(7): 73–78.

5. Chesbrough, H., and Spohrer, J. (2006) A Research Manifesto for Services Science, *Communications of the ACM*, 49(7): 35–40.
6. Clarke, R. J., and Nilsson A. G. (2007) Services Science from an IS Perspective: A Work Practice Approach for Analysing Service Encounters. In Ford, R. C., Dickson, D. R., Edvardsson, B., Brown, S. W., and Johnston, R. (eds) *Managing Magical Service, Proceedings of the QUIS 10 Conference*, June 14–17, Orlando, Florida, USA, pp. 54–62. The Rosen College of Hospitality Management, University of Central Florida.
7. Clarke, R. J., and Nilsson A. G. (2008) Business Services as Communication Patterns: A Work Practice Approach for Analysing Service Encounters, *IBM Systems Journal*, 47(1): 129–141.
8. Davenport, T. H. (1998) Putting the Enterprise into the Enterprise System, *Harvard Business Review*, July–August, 76(4): 121–131.
9. Davenport, T. H. (2000) *Mission Critical: Realizing the Promise of Enterprise Systems*, Boston, MA: Harvard Business School Press.
10. Davenport, T. H., Harris, J. E., and Cantrell, S. (2004) Enterprise Systems and Ongoing Process Change, *Business Process Management Journal*, 10(1): 16–26.
11. Demirkan, H., Kauffman, R. J., Vayghan, J. A., Fill, H.-G., Karagiannis, D., and Maglio, P. P. (2008) Service-Oriented Technology and Management: Perspectives on Research and Practice for the Coming Decade, *Electronic Commerce Research and Applications*, 7(4): 356–376.
12. Edvardsson, B., Gustafsson, A., and Enquist, B. (2007) Success Factors in New Service Development and Value Creation through Services. In Spath, D., and Fähnrich, K.-P. (eds), *Advances in Services Innovations*, pp. 166–183. New York: Springer.
13. Edvardsson, B., Gustafsson, A., Johnson, M. D., and Sandén, B. (2000) *New Service Development and Innovation in the New Economy*. Lund, Sweden: Studentlitteratur.
14. Glushko, R. J. (2008) Designing a Service Science Discipline with Discipline, *IBM Systems Journal*, 47(1): 15–27.
15. Hedman, J., and Lind, M. (2009) Is There Only One Systems Development Life Cycle? In Barry, C., Conboy, K., Lang, M., Wojtkowski, G., and Wojtkowski, W. (eds), *Information Systems Development: Challenges in Practice, Theory, and Education. Proceedings of the 16th International Conference on Information Systems Development – ISD’2007*, August, 29–31, 2007, Galway, Ireland, Vol. 1, pp. 105–116. New York: Springer.
16. Horn, P. (2005) The New Discipline of Services Science, *BusinessWeek*, January 21, 2005.
17. Klaus, H., Rosemann, M., and Gable, G. G. (2000) What is ERP? *Information Systems Frontiers*, 2(2): 141–162.
18. Larson, R. C. (2008) Service Science: At the Intersection of Management, Social, and Engineering Sciences, *IBM Systems Journal*, 47(1): 41–51.
19. Lusch, R. F., Vargo, S. L., and Wessels, G. (2008) Toward a Conceptual Foundation for Service Science: Contributions from Service-Dominant Logic, *IBM Systems Journal*, 47(1): 5–14.
20. Nilsson, A. G. (1990) Information Systems Development in an Application Package Environment. In Wrycza, S. (ed), *Proceedings of the Second International Conference on Information Systems Developers Workbench*, September 25–28, 1990, University of Gdansk, Poland, pp. 444–466.
21. Nilsson, A. G. (2001) Using Standard Application Packages in Organisations: Critical Success Factors. In Nilsson, A. G., and Pettersson, J. S. (eds), *On Methods for Systems Development in Professional Organisations: The Karlstad University Approach to Information Systems and Its Role in Society*, pp. 208–230. Lund, Sweden: Studentlitteratur.
22. Nilsson, A. G. (2007) Enterprise Information Systems – Eight Significant Conditions. In Knapp, G., Magyar, G., Wojtkowski, W., Wojtkowski, W. G., and Zupancic, J. (eds), *Information Systems Development: New Methods and Practices for the Networked Society, Proceedings of the 15th International Conference on Information Systems Development – ISD’2006*, August 31–September 2, 2006, Budapest, Hungary, Vol. 2, pp. 263–273. New York: Springer.
23. Nilsson, A. G. (2009) From Standard Application Packages to Enterprise Systems – A Matter of Opportunities. In Papadopoulos, G. A., Wojtkowski, W., Wojtkowski, G., Wrycza, S., and

- Zupancic, J. (eds), *Information Systems Development: Towards a Service Provision Society, Proceedings of the 17th International Conference on Information Systems Development, ISD'2008*, (pp. 443–440) August 25–27, 2008, Paphos, Cyprus. New York: Springer.
24. Orlikowski, W. J., and Iacono, C. S. (2001) Research Commentary: Desperately Seeking the 'IT' in IT Research: A Call to Theorizing the IT Artifact, *Information Systems Research*, 12(2): 121–134.
 25. Robey, D., and Markus M. L. (1998) Beyond Rigor and Relevance: Producing Consumable Research about Information Systems, *Information Resources Management Journal*, 11(1): 7–15.
 26. Simon, H. A. (1965) *The Shape of Automation for Men and Management*. New York: Harper and Row.
 27. Song, W., and Chen, D. (2009) An Examination on Service Science: A View from e-Service. In Papadopoulos, G. A., Wojtkowski, W., Wojtkowski, G., Wrycza, S., and Zupancic, J. (eds), *Information Systems Development: Towards a Service Provision Society, Proceedings of the 17th International Conference on Information Systems Development, ISD'2008*, (pp. 187–195) August 25–27, 2008, Paphos, Cyprus. New York: Springer.
 28. Spohrer, J., Maglio, P. P., Bailly, J., and Gruhl, D. (2007) Steps Toward a Science of Service Systems, IEEE Computer Society, January 2007, *Computer*, 40(1): 71–77.
 29. Sumner, M. (2005) *Enterprise Resource Planning*. Upper Saddle River, New Jersey: Pearson Prentice Hall.
 30. Sutton, S. G. (2006) Enterprise Systems and the Re-shaping of Accounting Systems: A Call for Research, *International Journal of Accounting Information Systems*, 7(1): 1–6.

A Petri Net-Based Software Process Model for Developing Process-Oriented Information Systems

Yu Li and Andreas Oberweis

Abstract Aiming at increasing flexibility, efficiency, effectiveness, and transparency of information processing and resource deployment in organizations to ensure customer satisfaction and high quality of products and services, process-oriented information systems (POIS) represent a promising realization form of computerized business information systems. Due to the complexity of POIS, explicit and specialized software process models are required to guide POIS development. In this chapter we characterize POIS with an architecture framework and present a Petri net-based software process model tailored for POIS development with consideration of organizational roles. As integrated parts of the software process model, we also introduce XML nets, a variant of high-level Petri nets as basic methodology for business processes modeling, and an XML net-based software toolset providing comprehensive functionalities for POIS development.

Keywords Process-oriented information systems · Software process model · Petri nets · XML nets

1 Introduction

The dynamics of the environment and the increasing competitive pressure compel traditional function- or task-oriented organizations to shift to process orientation in order to ensure customer satisfaction and high quality of their products and services. Process orientation indicates a way of thinking that regards organization's overall business activities as arranged business processes which logically link individual procedures or activities to collectively realize business objectives. It requires organizations to efficiently and flexibly design, model, control, analyze, execute, monitor,

Y. Li (✉)

Institute of Applied Informatics and Formal Description Methods (AIFB), Karlsruhe Institute of Technology (KIT), Universität Karlsruhe (TH), 76128 Karlsruhe, Germany
e-mail: yu.li@aifb.uni-karlsruhe.de

and constantly improve their business processes with the aid of process-oriented information systems.

Process-oriented information systems (POIS) represent a promising realization form of computerized business information systems. They aim at increasing flexibility, efficiency, effectiveness, and transparency of information processing and resource deployment in organizations by providing comprehensive and flexible automatization support for performing and controlling complex business tasks. Due to the complexity of POIS, explicit and specialized software process models are needed to guide POIS development. A software process model abstractly describes structure and properties of a class of processes that arrange activities required for developing, testing, and maintaining software products. A complete software process model should also provide methods and software tools to support performing software process activities [13].

Since the 1980s, a number of formalisms or languages have been proposed for modeling software development processes (see [1] for details), among which Petri nets have increasingly gained importance due to their formal semantics, graphical nature, high expressiveness, analyzability, and vendor-independence [23]. In this chapter we characterize POIS with an architecture framework and present a Petri net-based software process model specialized for POIS development with consideration of organizational roles. As integrated parts of the software process model, XML nets, a variant of high-level Petri nets as basic methodology for modeling business processes, and an XML net-based software toolset supporting POIS development will also be introduced.

The rest of the chapter is organized as follows: The next section surveys related work and discusses necessity and contributions of our work. In Section 3, POIS are characterized with an architecture framework. Section 4 describes the Petri net-based software process model for POIS development including the introduction of XML nets and the toolset. Section 5 concludes the chapter with an outlook on future research.

2 Related Work

In the literature a variety of (generic) software process models have been proposed, e.g., waterfall model [20], Spiral model [3], and V-model XT [6]. However, they mainly focus on general software development without taking into consideration the special characteristics of and requirements on POIS development. Moreover, they do not provide specialized methodologies for business process modeling and software tools to support POIS development.

In [10], a specialized software process model for workflow management applications is proposed. Based on a phase model for information systems development described in [17], it divides the process of developing workflow management applications into seven phases: enterprise planning, business domain analysis, system design reconstruction, system specification, module programming, system integration, and execution. The main problem of this model is that it does not consider the constant improvement of workflow management applications and the

business processes they support. It also does not provide any specific methods and tools.

On the other hand, there are several approaches for POIS development without providing explicit software process models. In [21], an approach named HOBE (House of Business Engineering) for developing and using POIS is described. It provides an architecture framework for business process management using standard application systems. The framework consists of four layers: process design layer, process planning and control layer, workflow control layer, and application systems layer. However, it does not take into consideration the organizational structure and resource deployment that are important for designing and operating business processes.

A similar approach DEM (Dynamic Enterprise Modeling) [19] also provides an architecture framework based on Baan application systems, which consists of four main components interacting with each other: Baan Enterprise Modeler, Baan Enterprise Workflow Management, Baan Enterprise Decision Support, and Baan/ Partner Applications. Although taking organization models into account, it still does not include specialized components for resource modeling and deployment.

3 Process-Oriented Information Systems

As preliminaries for understanding POIS, we first recall the concepts of information systems and business processes. In general, an *information system* (IS) can be regarded as a system of communication between people. It is involved in the gathering, storing, processing, distribution, disposition, and use of information [2, 14]. From a broader viewpoint, an information system might be regarded as a sociotechnical system consisting of people and machines that are interconnected through communication channels and create and/or use information with the aid of software techniques.

The Workflow Management Coalition (WFMC) defines a *business process* as “a set of one or more linked procedures or activities which collectively realise a business objective or policy goal, normally within the context of an organisational structure defining functional roles and relationships” [22]. A business process consists of manual, partially automated, or automated business activities executed according to given rules that prescribe the sequence (flow logic) of the process parts, activities, or functions. The computerized facilitation or automation of a business process, in whole or in part, is called a *workflow* [9]. The definition, control, execution, and monitoring of business processes or workflows are usually supported by *workflow management systems* (WfMS). To increase economic efficiency and cost effectiveness of the organization, it is necessary to constantly improve its business processes by means of business process reengineering.

A *process-oriented information system* (POIS), also called *process-aware information system* in the literature, can be defined as “a software system that manages and executes operational processes involving people, applications, and/or information sources on the basis of process models” [5]. A POIS usually comprises a WfMS as its core component to support process management and execution tasks.

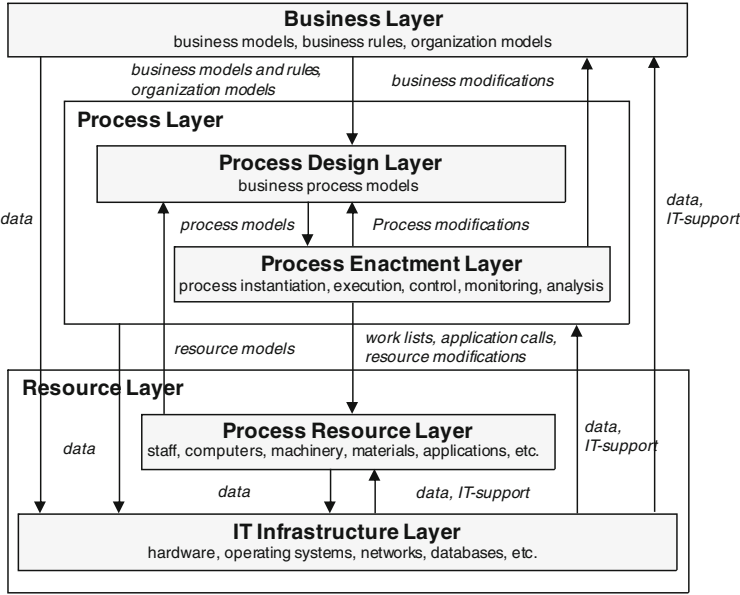


Fig. 1 Architecture framework of POIS

To better understand POIS, we provide an architecture framework (see Fig. 1) to describe the general architecture of computerized enterprise POIS. It consists of the following three layers:

1. *Business layer*: On this layer, business models, business rules, and organization models are analyzed and defined. A business model is a business concept containing value proposition, value creation architecture, and yield model of the organization. A business rule is used to direct or guide the organization’s business behavior. An organization model describes the structure of the organization in terms of responsibilities, roles, tasks, and relationships of the organizational units. Business models, business rules, and organization models are delivered as outputs of this layer to the process design layer.
2. *Process layer*: On this layer, business processes are modeled, executed, controlled, monitored, and analyzed. It is divided into two sublayers:
 - (a) *Process design layer*. This layer is devoted to delimitation and modeling of business processes. Inputs from business layer and process resource layer (resource models) are integrated into business process models. After having been validated (by simulation) and/or possibly formally verified, the resulting process models are sent as input to the process enactment layer.
 - (b) *Process enactment layer*. On this layer, business process models are instantiated and executed. In doing so, it allocates and uses the resources on the process resource layer by generating, assigning, and tracking work lists and

calling related applications or services. The process execution is controlled and monitored. The process models are, if possible, dynamically adjusted at run time. By analyzing execution data (based on given key indicators), necessary modifications to improve process, business, organization, or resource models are worked out and sent as suggestions to the process design layer, business layer, or process resource layer, respectively.

3. *Resource layer.* This layer covers all enterprise resources. It also consists of two sublayers:
 - (a) *Process resource layer.* This layer contains resources such as staff, computers, machinery, materials, and software systems, that are directly related to process execution. It defines and supplies resource models or specifications to the process design layer and executes work lists assigned by the process enactment layer.
 - (b) *IT infrastructure layer.* This layer contains IT infrastructure of the organization, e.g., hardware, operating systems, networks, and databases. It provides the other layers with IT support including networked communication and persistent data storage.

In comparison with the architecture frameworks mentioned in Section 2, our framework explicitly takes organization structure and resource deployment into consideration. Moreover, it provides an insight into how business process management and reengineering are supported by POIS.

4 Software Process Model for POIS Development

In this section we describe the software process model for POIS development. Petri nets are chosen as the modeling language due to their strengths such as formal semantics, mathematical foundation, high expressiveness, analyzability, executability, and support for hierarchical modeling. A Petri net is a directed, connected, and bipartite graph in which nodes represent places and transitions, and places can contain tokens. Variants of Petri nets can be classified in elementary and high-level Petri nets according to the used type of tokens. To ensure portability, we model the POIS development process using elementary Petri nets with the extension that allows the assignment of organizational roles to transitions.

In the following we first briefly introduce XML nets and the toolset as integrated parts of the software process model.

4.1 XML Nets

XML nets [15] represent a variant of high-level Petri nets, in which places are interpreted as containers for XML documents that must conform to XML Schemas

typing the places. Transitions can be inscribed by logic expressions, whose free variables must be contained in the inscriptions of adjacent edges, the so-called Filter Schemas used to read or manipulate XML data objects.

Figure 2 shows an XML net for simplified bill payment with XML Schema and Filter Schema diagrams assigned to places and edges, respectively. The XML Boolean element “isPaid” indicates whether the bill has been paid. The *transition inscription* “ $DA \geq 2009-1-1$ ” restricts that only the bills after January 1, 2009 are paid. The black bars in the Filter Schema diagrams represent *manipulation filters* used to create or delete XML documents in this context. The rectangles with an inscribed “A” stand for *element placeholders* of the XML data type *anyType* and can be instantiated by elements of any type. If the transition “pay bill” occurs, XML documents containing a bill number, a total price, an issue date, and an element “isPaid” with the value “false” are deleted from the place “bills unpaid” according to the Filter Schema assigned to the ingoing arc of the transition. According to the Filter Schema inscribing the outgoing arc of the transition and the XML Schema of the place “bills paid,” XML documents are created where the value of the element “isPaid” is set to “true.”

While retaining the advantages inherited from Petri nets, XML nets have additional strengths in the description of process objects and interorganizational exchange of standardized structured data (e.g., XML documents), which makes XML nets an appropriate language for modeling, for example, interorganizational business processes [16] and web service composition processes [4].

4.2 Toolset Supporting POIS Development

To facilitate POIS development, we are developing an XML net-based open source software toolset named INCOME2010¹ that offers comprehensive functionalities such as graphical and hierarchical process modeling, animated token game and automated simulation, workflow engine, organigram editor, process fragment library,

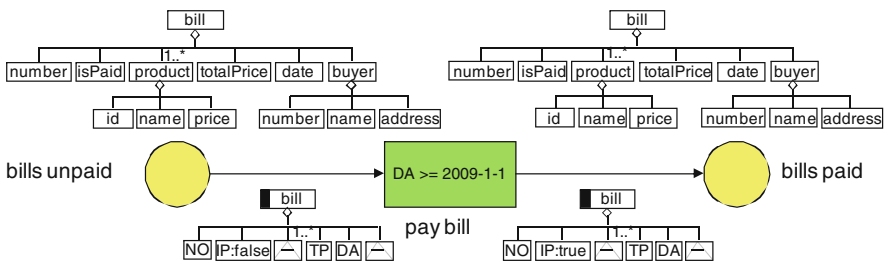


Fig. 2 XML net for paying bill

¹The current version of the toolset can be freely downloaded from the project’s website <http://www.aifb.uni-karlsruhe.de/Forschungsgruppen/BIK/income2010/>.

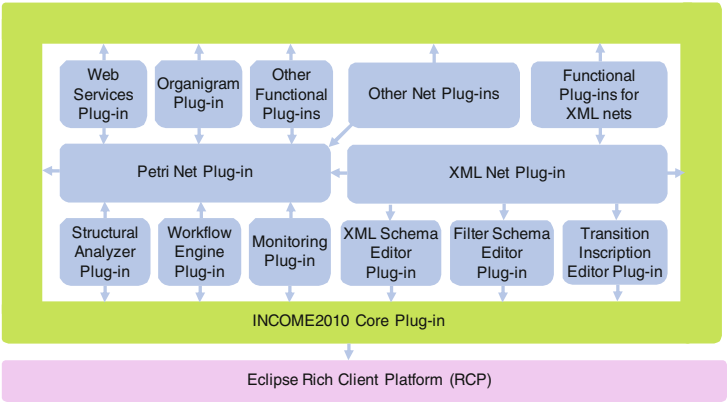


Fig. 3 Architecture of INCOME2010

web service interface, and definition and assignment of roles, resources, and process metrics (e.g., transition duration, costs, and place capacity).

INCOME2010 is a rich client Java application based on Eclipse Rich Client Platform (RCP). Taking advantage of the plug-in technology of Eclipse, it arranges its components in an open plug-in architecture in order to increase feature extensibility. Figure 3 depicts the current architecture of INCOME2010. More details about this toolset are found in [12].

4.3 Software Process Model

Figure 4 depicts an overview of the software process model consisting of five phases: planning, analysis, design, operation, and maintenance. Each phase is realized by a corresponding transition that is further refined by a subdiagram containing details of the phase. The places in the Petri net model represent containers of data or documents as inputs or outputs of the phases/transitions. The place “project analysis data” is the central storage of analysis results of all phases used for improvement

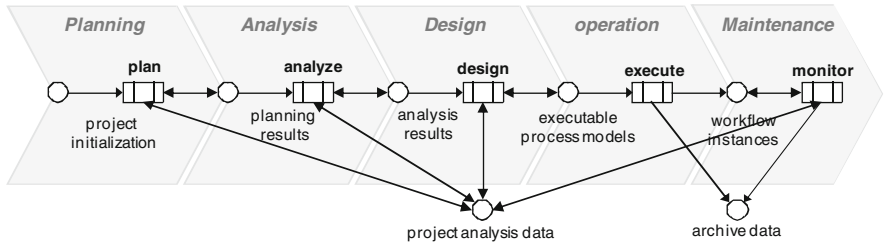


Fig. 4 Overview of the software process model for POIS development

or reconfiguration purpose. The place “archive data” stores discarded and archived data such as process interpretations, workflow instances, and execution protocols. The phase implementation is omitted here, as the suggested toolset INCOME2010 already provides workflow management functionalities such as process design, execution, control, and monitoring. Activities of the test phase are integrated as process model validation into the phase design.

4.3.1 Planning

The transition “plan” is refined by a subdiagram (see Fig. 5) describing the phase planning, in which project planner and customer work out together business process delimitation, project plan, and system goals and strategies. The planning results are checked by project manager and/or quality supervisor. If necessary, a replanning can occur to improve the planning results. Otherwise the next phase is triggered. In the diagram, the places framed with boxes stand for border places. Methods used in this phase could be SWOT [8], PESTLE [7], balanced scorecard [11], etc. Supporting project planning tools may include Microsoft Project,² IBM Rational Tools,³ and SmartWorks Project Planner.⁴

4.3.2 Analysis

Figure 6 shows details of the phase analysis, in which analysts check the planning results and elaborate requirements catalog, key indicator system, risk assessments, and descriptive models including business, process, data, resource and organization

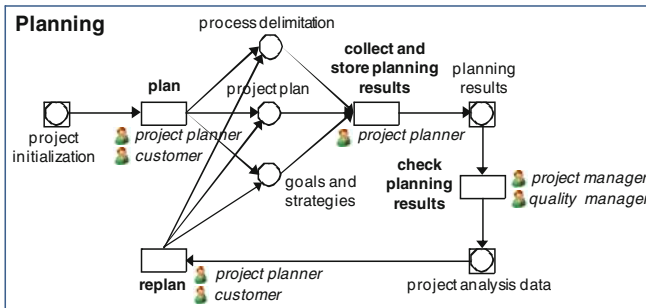


Fig. 5 Planning phase

²<http://office.microsoft.com/en-us/project/default.aspx>.

³<http://www-01.ibm.com/software/rational/>.

⁴<http://www.smartworks.us/htm/project.htm>.

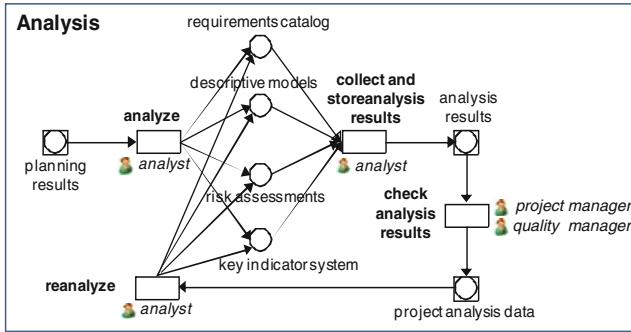


Fig. 6 Analysis phase

models. After having been checked by the project manager and/or the quality supervisor, the analysis results can be improved by replanning, or used directly in the next phase. Various analysis methods can be used in this phase, e.g., interviewing, questionnaires, use cases, prototyping, UML, and FRAP [18]. Tools supporting this phase may include ARM tool,⁵ RAVEN,⁶ Eclipse Uml2Tools,⁷ and PillarOne.⁸

4.3.3 Design

Based on the analysis results, designers develop in this phase (see Fig. 7) executable process models, in which business, organization and resource models are integrated. These process models are then validated or verified by process designers, project manager and/or quality manager. The models are redesigned if semantic faults or syntactic errors are found. For this phase, we suggest again XML nets as modeling language and INCOME2010 as supporting tool.

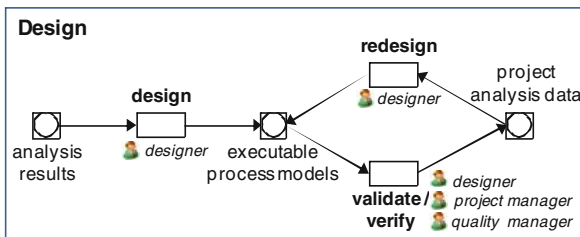


Fig. 7 Design phase

⁵<http://satc.gsfc.nasa.gov/tools/arm/>.

⁶<http://www.ravenflow.com/>.

⁷<http://www.eclipse.org/modeling/mdt/?project=uml2tools>.

⁸<http://www.pillarone.org/>.

4.3.4 Operation

In this phase (see Fig. 8), executable process models are first interpreted or parsed through a WfMS. Based on a process interpretation, the WfMS can generate several workflow instances for the execution, during which resources are allocated and relevant software applications or services are called. The process administrator can discard and archive (old) process interpretations to allow generating workflow instances on the basis of new interpretations, or to prevent generating too many workflow instances. The tool suggested for this phase is INCOME2010 providing an integrated XML net-based WfMS.

4.3.5 Maintenance

As shown in Fig. 9, in this phase the execution of workflow instances is monitored through WfMS. The execution data is stored as protocols that can be analyzed by process analysts. Based on the analysis results, the process administrator can reconfigure workflow execution at run time if necessary. The analysis results may also trigger improvements in other phases. Old, faulty, or meaningless workflow instances or protocols can be discarded and archived by the administrator. Methodologically, the analysis of execution data can be performed based on

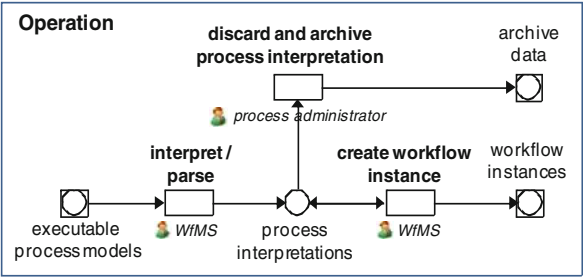


Fig. 8 Operation phase

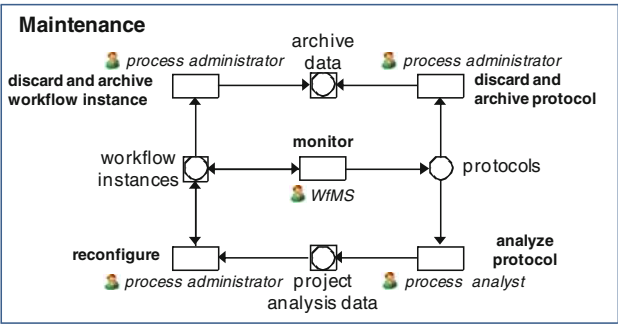


Fig. 9 Maintenance phase

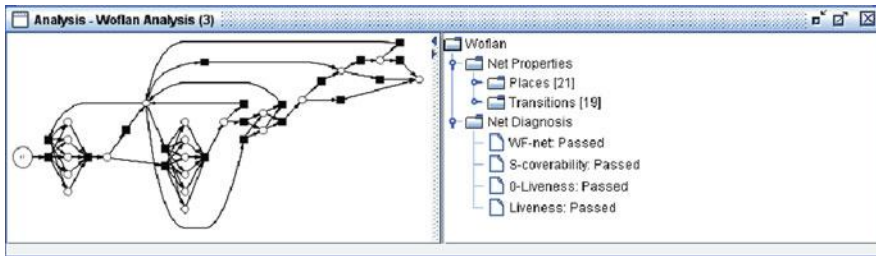


Fig. 10 Structure properties of the software process model

predefined key indicators (e.g., throughput, costs, and resource utilization) for the evaluation of performance, profitability, productivity, quality of service, etc. The suitable tools used to support this phase would be INCOME2010 and ProM.⁹

Using INCOME2010, the presented software process model has been validated through simulation. Its structure properties have also been verified: The model is formally unbounded, i.e., it contains unbounded places “workflow instances” and “protocols.” The reason is that from our viewpoint it is more close to the reality of workflow execution to allow the creation of several workflow instances based on one process interpretation, and several protocols based on one workflow instance. Nevertheless, to achieve formal boundedness, the model only needs to be slightly modified by deleting three arcs, i.e., from “create workflow instance” to “process interpretations,” from “monitor” to “workflow instances,” and from “workflow instances” to “reconfigure.” As verified with the Woflan analyzer provided by ProM (see Fig. 10), the modified model is a live, S-coverable, and sound workflow net.

5 Conclusions

In this chapter we characterized POIS with an architecture framework that takes into account organizational structure and resource deployment. To guide the development of complex POIS, we presented a Petri net-based software process model with XML nets as business process modeling language and an open source toolset INCOME2010 supporting the design, operation, and maintenance of POIS. Its advantages can be summarized as follows:

- It considers the general life cycle of IS development. A typical system development life cycle contains phases such as planning, analysis, design, implementation, testing, operation, and maintenance.
- It also takes into consideration specific properties of process orientation, i.e., it supports the elicitation, definition, modeling, execution, control, monitoring, analysis, and improvement of business processes.

⁹<http://prom.win.tue.nl/tools/prom/>.

- It is iterative to support business process reengineering and continuous improvement of business processes and related models.
- It reflects data and document flows in POIS development.
- It considers organizational roles participating in POIS development.
- Although tailored for POIS development, it is portable, i.e., it is not bound to specific POIS or supported business processes.
- It is hierarchical to provide different views with different levels of detail.
- As a Petri net model, it is executable so that it can be validated through simulation, and the modeled POIS development process can be automated.

Currently, we are evaluating the software process model in practical POIS development projects, while XML nets and INCOME2010 are already being applied in diverse industrial and research projects. Besides, our future work also includes extending INCOME2010 with functionalities for the phases planning and analysis.

Acknowledgment The authors would like to thank the anonymous referees for many valuable comments on an earlier version of this chapter.

References

1. Acuña, S. T., Antonio, A. de, Ferré, X., López, M., and Maté L. (2001) The Software Process: Modelling, Evaluation and Improvement. In Chang, S. K. (ed), *Handbook of Software Engineering and Knowledge Engineering*, World Scientific, New Jersey, EE.UU., pp. 193–237.
2. Beynon-Davies, P. (2002) *Information Systems: An Introduction to Informatics in Organisations*, Palgrave, New York, USA.
3. Boehm, B. W. (1988) A spiral model of software development and enhancement. *Computer*, 21(5): 61–72.
4. Che, H., Li, Y., Oberweis, A., and Stucky, W. (2009) Web Service Composition Based on XML Nets. In Ralph, H., and Sprague, J. (eds), *Proceedings of the 42nd Annual Hawaii International Conference on System Sciences (HICSS)*. IEEE, Hawaii, USA.
5. Dumas, M., van der Aalst, W. M. P., and ter Hofstede, A. H. M., (eds) (2005) *Process-Aware Information Systems: Bridging People and Software through Process Technology*, John Wiley & Sons, New Jersey, USA.
6. Federal Republic of Germany (2006) *V-Modell XT*, Version 1.3, <http://v-modell.iabg.de/dmdocuments/V-Modell-XT-Gesamt-Englisch-V1.3.pdf>.
7. Gillespie, A. (2007) *Foundations of Economics*, Oxford University Press, New York, USA.
8. Hill, T., and Westbrook, R. (1997) SWOT Analysis: It's Time for a Product Recall, *Long Range Planning* 30 (1): 46–52.
9. Hollingsworth, D. (1995) *The Workflow Reference Model*, Workflow Management Coalition, Document Number TC00-1003, Winchester.
10. Jablonski, S., and Stein, K. (1998) Ein Vorgehensmodell für Workflow-Management-Anwendungen (in German). In Kneuper, R., Müller-Luschnat, G., and Oberweis, A. (eds), *Vorgehensmodelle für die Betriebliche Anwendungsentwicklung*, B.G. Teubner, Stuttgart/Leipzig, Germany, pp. 136–151.
11. Kaplan, R. S., and Norton, D. P. (1992) The Balanced Scorecard – Measures that Drive Performance, *Harvard Business Review*, January–February, 71–80.
12. Klink, S., Li, Y., and Oberweis, A. (2008) INCOME2010 – A Toolset for Developing Process-Oriented Information Systems Based on Petri Nets. In *Proceedings of International Workshop on Petri Nets Tools and Applications*. ACM digital library, Marseille, France.

13. Kneuper, R. (1998) Requirements on Software Process Technology from the Viewpoint of Commercial Software Development: Recommendations for Research Directions. In Gruhn, V. (ed), *Proceedings of the 6th European Workshop on Software Process Technology*, Weybridge, UK, LNCS 1487, Springer.
14. Langefors, B. (1973) *Theoretical Analysis of Information Systems*, 4th edition. Studentlitteratur, Lund, Sweden.
15. Lenz, K., and Oberweis, A. (2003) Inter-Organizational Business Process Management with XML Nets. In Ehrig, H., Reisig, W., Rozenberg, G., and Weber, H. (eds), *Petri Net Technology for Communication Based Systems*, LNCS 2472, Springer, pp. 243–263.
16. Li, Y. (2007) *Umsetzung unternehmensübergreifender Geschäftsprozesse mit XML-Netzen* (in German), VDM Verlag, Saarbrücken, Germany.
17. Martin, J. (1990) *Information Engineering*, Englewood Cliffs, Prentice-Hall, New Jersey, USA.
18. Peltier, T. R. (2001) *Information Security Risk Analysis*, Auerbach.
19. Rodim van Es, M. (1998) *Dynamic Enterprise Innovation – Establishing Continuous Improvement in Business*, Netherlands: Baan Business Innovation, Modderkockl Ede.
20. Royce, W. W. (1987) Managing the development of large software systems: concepts and techniques. In *ICSE '87: Proceedings of the 9th International Conference on Software Engineering*, Los Alamitos, CA, USA. IEEE Computer Society Press, pp. 328–338.
21. Scheer, A.-W. (1998) ARIS – House of Business Engineering. In Molina, A., Kusiaka, A., and Sanchez, J. (eds), *Handbook of Life Cycle Engineering: Concepts, Models and Technologies*, Kluwer Academic Publishers, Dordrecht/Boston/London, pp. 331–357.
22. The Workflow Management Coalition, (1999) *Terminology & Glossary*, document no. WFMC-TC-1011.
23. Van der Aalst, W.M.P. (1998) The Application of Petri Nets to Workflow Management, *Journal of Circuits, Systems and Computers*, 8(1): 21–66.

Modern Enterprise Systems as Enablers of Agile Development

Odd Fredriksson and Lennart Ljung

Abstract Traditional ES technology and traditional project management methods are supporting and matching each other. But they are not supporting the critical success conditions for ES development in an effective way. Although the findings from one case study of a successful modern ES change project is not strong empirical evidence, we carefully propose that the new modern ES technology is supporting and matching agile project management methods. In other words, it provides the required flexibility which makes it possible to put into practice the agile way of running projects, both for the system supplier and for the customer. In addition, we propose that the combination of modern ES technology and agile project management methods are more appropriate for supporting the realization of critical success conditions for ES development. The main purpose of this chapter is to compare critical success conditions for modern enterprise systems development projects with critical success conditions for agile information systems development projects.

Keywords Enterprise systems · Success conditions · Project management · Agile methods · Web-based enterprise systems · Sales portals · Sales order processes

1 Introduction

1.1 Enterprise Systems

Today, there is a booming market for software packages claiming to provide a total, integrated solution to firms' information-processing needs. Enterprise systems (ESs) are the new type of configurable computer-based information systems for enterprise integration. ESs are sold as comprehensive software solutions that help to integrate business processes through shared information and data flows [26]. The integration

O. Fredriksson (✉)

Department of Information Systems, Karlstad University, Karlstad, Sweden
e-mail: odd.fredriksson@kau.se

of core business functions, including order management and logistics, is achieved through creation of a single system with a shared database [18].

Over the past 15 years, many large and SME organizations have adopted ESs expecting positive outcomes. Many of these firms are now upgrading, replacing, or extending their original ESs. The development and implementation of these large integrated systems represent major technical and organizational challenges.

Because traditional ESs are complex and rigid, they are incapable of boosting higher productivity [9]. The reason is argued to be that the traditional ESs are monoliths with complicated code bases, which make it difficult to design solutions meeting the needs of the individual firm. It is not possible either to just pick one improvement that the customer wants from an upgrade and skip the other. Instead it is all or nothing.

Vendors of traditional ESs are aware of the growing dissatisfaction among their users who are forced to work in cumbersome and inefficient ways, hire expensive consultants to simply manage the systems, and perform Big Bang upgrades causing lots of disruption and cost.

1.2 A Paradigm Shift From Tightly to Loosely Coupled Enterprise Systems

The current movement to Service Orientation is indicating an era of enterprise computing based on open standards, Service-Oriented Architecture (SOA), and optimized business processes. SOA is a style of enterprise architecture that enables the creation of applications and processes, built by combining interoperable services in a loosely coupled fashion. These services interoperate based on a formal definition, independent from the underlying resources and programming language. SOA offers more flexibility, lower costs and increased productivity [7], and enables the firm to make itself free from hard coded, heterogeneous, difficult-to-integrate and fragmented applications [3]. This indicates a radical shift from today's rigid and inflexible systems towards loosely coupled approaches to ES application development and diffusion. Functionality thereby becomes redefined so that capability is accompanied with usability and flexibility.

ComActivity, an ES vendor firm based in Stockholm, Sweden, with about 40 consultants, offers solutions belonging to this new modern ES generation, based on open standards such as Java, Eclipse, Web 2.0, BPM and Model-Based Applications (MBA). ComActivity's graphic modelling, instead of coding when designing lean and flexible business processes, eliminates 90% of the code necessary to represent a business process. One illustration of the difference in level of complexity is that the SAP system consists of about 45,000 tables while the ComActivity system consists of about 2,500 tables [2]. ComActivity's method is that new solutions should be implemented step by step in relatively limited projects which create business value fairly immediately.

1.3 Project Management Perspective

Traditional project management methods originate from the large development projects carried out during the 1940s–1950s within the American military industry [12]. These traditional methods feature extensive upfront planning as the basis for predictions, measurements and control during the project life cycle [21].

Nowadays, most projects are still managed by the same principles using the same methods, usually without further reflection, even in small projects of short duration [23]. Projects are generally defined as non-recurrent assignments with well-defined goals, time limit and budget [19], but there is a wide range of project types requiring a wide range of management methods [23]. Extensive research has proved that traditional project planning and management methods are insufficient for project success within information systems development [12, 23]. To exemplify, The Standish Group International [31] found that not more than about one fourth (28%) of 280,000 surveyed new-start application software development projects were perceived as successful. As Chow and Cao [6: 961] posit:

Despite the efforts to employ software engineering methodologies, software development has not been consistently successful, thus often resulting in delayed, failed, abandoned, rejected software projects.

One pressing challenge therefore is to explore and reflect on how to improve software development project management. Goldsmith [15] argues that one overlooked major real cause of IT project failures is that management arbitrarily dictates project budgets and schedules “which bears no relationship to the work to be done . . . When budgets and schedules are nonsense, overrunning them becomes nonsense too.”

In The Standish Group International study [31], the strength of influences from different success factors on a project’s success was also surveyed. Following closely after “Executive support” and “User involvement”, the third most important success factor was an “experienced project manager”. Of the successful projects 97% had an experienced project manager.

In this chapter we focus mainly on ES development. We acknowledge, however, that successful ES change does not only require successful ES development, but also successful IT implementation. As Mathiassen and Pries-Heje [20: 117] put it:

Numerous are the IT projects that succeeded in developing a product but failed in changing the behaviour of the target group. Diffusion just did not happen. Therefore, agile IT diffusion is sought-after.

1.4 Agile Project Management – A New Paradigm?

There is a growing popularity of agile methods within project management [21]. Instead of extensive upfront project planning, project plans are made mainly for flexibility and changes, and the purpose of project evaluation is not to compare the progress with the original plan, but to decide new roads of action for the project.

Agile methods are characterized by short iterative cycles of development guided by product features and continuous integration of code changes during development. The deliverables from each development cycle is working code that can be used by the customer. Agile development emphasizes “the people factor” [6]. Nerur et al. [21] summarize that agile development is characterized by both system developers and end-users playing important roles and that they form teams that collaborate in repeated cycles of thought–action–reflection. Team roles are flexible and the communication is informal.

1.5 Critical Success Factors

The Critical Success Factor (CSF) approach was originally suggested by Rockart [24]. A CSF indicates what has to be managed correctly to achieve values, positive outcomes or successful results.

We need to develop a better understanding of how, why and when values, or positive outcomes, can – or cannot – be generated from IT use [28, 32]. These researchers advocate process theory models, which contain arguments of the type “necessary, but not sufficient” conditions to realize effects as a result of the adoption and use of an IT artefact. Or, expressed differently, a process theory maintains that the outcome will only occur under the circumstances specified by the theory. At the same time a process theory states that the outcome may not occur even if all the specified prerequisites, or conditions, are in place [21]. Acknowledging the dynamic contexts for ES development and implementation processes, in this chapter we therefore use the process theory concept “condition” instead of the variance theory concept “factor”.

1.6 Purpose

The main purpose of this chapter is to compare critical success conditions for modern enterprise systems development projects with critical success conditions for agile information systems development projects. This comparison is made mainly from a customer perspective. By “customer perspective” it is meant that the main focus is on the organization using the enterprise system.

1.7 Method

Both study objects in the purpose chosen for this chapter (modern enterprise systems and agile system development methods) are recent phenomena, which motivate exploratory research approaches when studying these phenomena. The research design applied in this chapter is to contrast an explorative case study of a change

from a traditional ES to a modern ES, after a successful ES development process [14], in relation to findings from a specific explorative survey study on critical success factors of agile software development projects [5]. The authors of this chapter belong to two different academic disciplines, which enable application and combination of the respective perspectives on the problem area.

One of the authors conducted the case study of the successful ES change, which is very briefly presented below. Mainly personal interviews and attending project meetings were used as methods for collecting empirical data. Personal interviews were made with top managers, one super user, order assistants, one logistics manager, finance assistants, and with ES vendor representatives. The definition of super user applied in this chapter is: the co-worker being most knowledgeable about the focal case firm's sales order processes. The case study was conducted from early 2006 until June 2008. A condensed form of the case study has been published as a book chapter [14].

The working procedure for "the management of agile information systems development projects" part of this chapter was to find a suitable research literature-based research framework. We found the research model of Chow and Cao [5: 964] on success factors of agile system development projects to be relevant, useful and simple. Therefore, in the analysis and discussion part of this chapter we relate our collected empirical data to the findings made by Chow and Cao [5].

2 The Focal Case Study Firm

In 2007, the chosen focal firm for the case study presented here, that is Wermland Paper (WP), chose to change ES from Movex (now renamed to Lawson M3) to ComActivity. WP is a sack paper producer, which holds market-leading positions within selected niches for unbleached craft paper, both in Europe and globally. It has a 90% export share to customers in about 70 countries. It has a fragmented customer structure with 800 customers. Its annual turnover in 2007 was about 1.1 billion SEK and the number of employees was around 360. This middle-sized case firm has two paper mills, located in the county of Wermland, in West Sweden.

WP was formed in 2003 by a merger between the Bäckhammar and Åmotfors mills, when a venture capital firm and the new top management team bought the family firm which was established in the nineteenth century. It was concluded that in order for WP to survive in evolving new market conditions, a clear strategy with IT implementation was needed. Since 2008, WP is part of the Nordic Paper group.

2.1 Main Steps in the Development Project of the New ES

(1) Redesign to a common sales order process for the two mills; (2) choice of ES vendor to conduct a pilot study; (3) acceptance-building meetings for the planned

customer portal with co-workers and agents; (4) the board's decision to purchase a Web-based ES (modules for order, inventory and invoicing) from ComActivity; and (5) development phase with the main ComActivity developer and two super users.

Because of the 12-page restriction, the description of what happened during the main steps in this development project had to be excluded from this chapter. The main findings are very briefly reported later in this chapter.

2.2 Some of the Initial Effects for the Case Study Firm

In January 2007, the new Web-based order function, or Sales Portal, was opened for the internal users. In April 2007, this Sales Portal was opened for the agents. In September 2007, five of its largest agents pre-booked 35% of WP's total sales order volume. In March 2008, 40% of WP's total sales order volume was pre-booked by its larger agents. One year after WP's new ES "went live", the development process of this modern ES and the following implementation process both were perceived by the different stakeholders to have been successful.

2.2.1 Business Impacts From ES Use After 1 Year

New functionality: the agents can visually access information that they need via the Sales Portal all around the clock; improved sales order process, which takes less time and resources for WP; less errors since orders are placed only once; full traceability of data when customer complaints are made; and correct inventory data.

2.2.2 Organizational Impacts From ES Use After 1 Year

All sales order information is now on one screen, compared to 12 screens before the ES change; it is much easier to learn the new ES; and since the new ES is much simpler to use, there are many more co-workers at WP who are placing sales orders than before. The order assistant role is taken away; however, the market assistants cannot be sure that the agents have pre-booked their orders correctly via the Sales Portal. They still have to check the orders. They also have to contact production planning, just as before the ES change, for setting delivery dates.

3 Findings on Critical Success Conditions From the Case Study

The profit margins improved significantly for the focal case firm over the 2003–2007 period. In 2007, its profit margin exceeded 10%. The CEO of WP comments that "a large proportion of WP's growth is attributable to the integration and dialogue with our customers. Of WP's turnover 70% consists of sales to customers to which WP is the dominant seller". Thus, the Sales Portal – offered to its agents in spring 2007 – is an important component in WP's current strategy.

In this part of the chapter, some of the inductive conclusions drawn from the WP case study are presented. Practitioners in managerial positions can reflect on these when a business-critical system such as a new ES is going to be selected, developed and implemented. What are the main conditions explaining the successful ES change process of the studied case firm? Each one of the following four main critical success conditions is argued to a high degree explain the perceived success for the case firm:

1. The top management's experiences from ES changes (CSC 1).
2. The right basic ES selection decision (CSC 2).
3. A good symbiosis between competent system developers and super users (CSC 3).
4. A short ES change project with a high pace (CSC 4).

3.1 Top Management's Experiences From ES Changes (CSC 1)

The three "heaviest" ambassadors in the top management team – the CEO, the marketing director and the IT director – all had experiences from ES change processes which had turned into failures. In other words, the top management had competence about the challenges and pitfalls associated with ES change processes. Therefore, relatively heavy argumentation from the IT director was required in order to convince the CEO and the marketing director about the necessity to make the decision to change ES modules.

The positive side of inertia in organizations is that it makes you think and reflect. The top management team wanted to be sure about a successful ES change process when it made its decision. The insights of the three ambassadors in terms of knowledge about the history and about current ESs, their "bad" experiences, and their commitment and support altogether was an important condition explaining the success of this ES change project.

3.2 The Right Basic ES Selection Decision (CSC 2)

The decision-makers at WP perceived that the ComActivity system had the actual capability to deliver what the users actually demanded. Thus, the selected ES was the right "product" for the focal case study firm. The flexibility and simplicity associated with this modern ES enabled the actual end-users to have the system designed as they wanted it to be. This was a very important critical success condition. Thus, the classical misalignment between the business processes and the ES package (see, e.g. [8, 27, 29]) was not an issue in this project. The vendor (ComActivity) therefore argues that its ES means a paradigm shift, as the customer with simple means can obtain optimized and customized systems, that supports the business of the customer, neither more nor less [2].

3.3 A Good Symbiosis Between System Developers and Super Users (CSC 3)

The IT director of WP was full of faith when it came to choosing the single most important success condition for successful ES changes: “It is enormously important to have the right ES consultant or consultants”. To engage wrong consultant, who cannot speak with the end-users, can be disastrous. Already back in 1983, Block [4] identified inability to communicate with the system users as one of the cause categories for enterprise project failures. The fact that the main ComActivity developer spoke a language that the super users and end-users at WP could follow was a central success condition.

The high competence of the super users was of great importance for the success of the ES project. Also, the active participation of the market assistants (i.e. end-users of the order system) in the project, who earlier had been critical towards the ES, was another important success condition (cf. [16]). In addition, it is required that the sum of the individuals function as a team, that is a good project team. There was a good symbiosis in place between highly competent system developers and super users in this ES change project. Such everyday professional cooperation between the participating individuals is obtained if the project members work well together and “swim in the same direction”.

3.4 A Short ES Change Project With a High Pace (CSC 4)

Willcocks and Sykes [33: 37] have stressed the importance of defining focused and short ES projects. To run a project with a high pace during a long time period is of course trying for all those involved. To run a long project with a low pace makes it long-winded, and tiredness emerges and setbacks appear. Thus, it is highly essential that an ES project is short and has a high pace. As the ES development project at WP lasted for 4 months it was not exaggeratedly long.

It is very important that projects reach their milestones and progress. It is psychologically important that the end-users perceive that something is happening and that both they and the system consultants experience that the change unfolds. If the end-users perceive that the project is long-winded, they start to despair and believe that the project never will come in. Since the pace was high in this ES development project, this feeling never appeared. The IT director is convinced that all involved in the ES change project would agree with that. However, the pace should not be so high that there is no time for reflection.

A fundamental prerequisite for being able to have a high pace in an ES change project is that the super users to a high degree are relieved from their ordinary work tasks. The top management team and the super users agreed that this was appropriate. The super users only had some overtime during the project.

In the case firm it was mainly one department, the sales order department, involved in the order, inventory and invoicing ES change project. But when required, the finance and logistics departments were also involved. If the project had been larger with the same project group, large project size problems probably would have

emerged. Some of the reasons for these problems are that complexity increases, the project becomes long and vacations interfere.

4 Critical Success Conditions for Agile ISD Projects

Chow and Cao [5] made an explorative study to find out which factors can positively influence on the success of agile system development projects. The group coordinators for 109 agile system development projects, from 25 countries, responded to their survey, which was based on the success factors reported in “the agile literature”. After factor analyses, they identified 12 success factors, which were classified into five categories, or dimensions: Organizational, People, Process, Technical and Project.

The *Organizational dimension* is formed by conditions such as strong management commitment, cooperative organizational culture and team environment.

The *People dimension* is formed by conditions such as “team members with high competence” and strong customer involvement.

The *Process dimension* is formed by conditions such as following agile-oriented management process and strong communication focus with daily face-to-face meetings. By working agile-oriented means, e.g. that the project team is not tied up to original requirements, original project plans or specified timetables.

The *Technical dimension* is formed by conditions such as “pursuing simple design”, “delivering most important features first” and regular delivery of software (forming the “delivery strategy” condition).

The *Project dimension* is formed by conditions such as “project type being of variable scope with emergent requirement” and “projects with small teams”.

Reformulated into process theory terms, the research framework says that these are “necessary, but not sufficient” conditions to realize “Perceived Success” effects as a result of successful agile system development projects.

The success conditions found to be significant in the empirical survey of Chow and Cao [5] were the following (in descending ranking order of importance): delivery strategy and agile software engineering techniques (technical), team capability (people), project management process (process), team environment (organizational) and customer involvement (people).

5 Analysis and Discussion

In the concluding part of this chapter, we will relate findings from the case study of Wermland Paper to some of the critical success conditions identified by Chow and Cao [5], structured along their five dimensions.

5.1 Organizational Dimension

The importance of the top management commitment and support condition has been demonstrated in many studies. Duchessi et al. [11: 78], for example, in their survey

study found that more support is given from top management teams in organizations which successfully implement ESs than what is given in organizations which have been less successful. The “ambassadors” became part-owners of the case firm when they were recruited, which very likely strengthened their management commitment (CSC 1).

If the top management team is not aware of how to avoid the pitfalls, then things can go really wrong. Along the same vein, Sumner [30: 115] advises top managers to aim for reducing the risks associated with ES implementation projects. This requires that the top management is capable of identifying the risks and can assess their amplitudes. The top management of the case firm had competence about the challenges and pitfalls associated with ES change processes (CSC 1).

5.2 People Dimension

The case firm project team conducted its development work according to agile method characteristics (cf. [22]). A small team with highly competent team members was picked: one ES developer/consultant in the forefront (with two developers back-front), two super users (representing system users) and one ambassador (with two more top managers back-front). The IT director of the case firm is convinced that “the right ES consultant/s” is the single most important success condition (CSC 3). Conditions such as the high competence of the super users and the active participation of the market assistants (i.e. end-users of the sales order system) in the project were also of great importance for the success of the ES change project (CSC 3).

5.3 Process Dimension

Communication failure is argued to be one major cause for IS project failures according to Sumner [30]. This major risk was offset by the third critical success condition (CSC 3) for the case firm. The main ComActivity developer applied a people-centric approach for the development work: short, iterative, test-driven processes that encouraged feedback from the super users and also from the end-users. User contact failure risks, such as ineffective user communication and lack of user commitment [17], were offset by the communication skills competence of the main ComActivity developer (CSC 3).

Feeny and Willcocks [13] have stressed the importance of relationship building to establish understanding, trust, and cooperation among the end-users and the IT personnel. The full potential from an ES cannot be obtained without a strong coordination of effort and goals across business and IT personnel. As Daghfous [10] argues, learning alliances is a fast and effective mechanism of capability development. This was an important precondition for the user-driven processes to become successful and to result in positive outcomes.

5.4 Technical Dimension

The new modern ES technology enables for the project team to work agile-oriented during ES development. The new modern ES technology also enables flexibility in terms of allowing the project team to dynamically develop the ES in an iterative way, markedly improving the possibility to achieve high customer value and high usability. Modern ES technology matches what Nerur et al. [22: 75] claim to be a fundamental assumption for the path of agility:

..., adaptive software which can be developed by small teams using the principles of continuous design improvement and testing based on rapid feedback and change.

In the Chow and Cao [5] study, the technical dimension (agile software techniques and delivery strategy) was found to be the most critical dimension in impacting the success of agile projects.

Technology failure risks such as failure of the information system to meet specifications was offset in the case firm by the flexibility of the new modern ES (CSC 2) and the highly competent main ComActivity developer (CSC 3).

5.5 Project Dimension

The positive benefits associated with settling many milestones and meeting these before reaching the goal have been stressed by many (see, for example, [25]; CSC 4). The high failure risk associated with large projects [1] was handled by the case firm through choosing a “small step, small win” development and implementation strategy (CSC 4), which enabled the ES change project to become short.

6 Conclusions

Traditional ES technology and traditional project management methods are supporting and matching each other. But they are not supporting the critical success conditions for ES development in an effective way. Although the findings from *one* case study of a successful modern ES change project is not strong empirical evidence, we carefully propose that the new modern ES technology is supporting and matching agile project management methods. In other words, it provides the required flexibility which makes it possible to put into practice the agile way of running projects, both for the system supplier and for the customer. This conclusion is in line with what Nerur et al. [22: 77] posit to be a fundamental condition:

Tools play a critical role in successful implementation of a software development methodology. Organizations planning to adopt agile methodologies must invest in tools that support and facilitate rapid iterative development.

In addition, we propose that the combination of modern ES technology and agile project management methods is more appropriate for supporting the realization of critical success conditions for ES development.

This chapter represents an attempt at bridging ES practice and agile ISD literature. One suggested research questions for future research following from this endeavour is: How will new ES technology and agile project management methods change the established critical success conditions for ES development projects?

References

1. Barki, H., Rivard, S., and Talbot, J. (1993) Toward an Assessment of Software Development Risk, *Journal of Management Information Systems*, 10(2): 203–225.
2. Björkman, P. (2008) CEO ComActivity. Personal conversation on January 14, Stockholm, and telephone conversation on May 18.
3. Björkman, P. (2008) SOA – idag och imorgon, presentation at Computer Sweden's SOA Summit 2008. Retrieved August 28, 2008 from: <http://www.comactivity.net/downloads/files/ComActivity%20Enables%20Your%20SOA.pdf> (in Swedish).
4. Block, R. (1983) *The Politics of Projects*. Yourdon Press, Prentice-Hall, Englewood Cliffs.
5. Chow, T., and Cao, D.-B. (2008) A Survey Study of Critical Success Factors in Agile Software Projects, *The Journal of Systems and Software*, 81: 961–971.
6. Cockburn, A., and Highsmith, J. (2001) Agile Software Development: The People Factor, *IEEE Computer*, 34(11): 131–133.
7. ComActivity (2008) ComActivity Enables Your Service Oriented Architecture. Retrieved March 13, 2008 from: <http://www.comactivity.net/downloads/files/ComActivity%20Enables%20Your%20SOA.pdf>.
8. Computer Sweden (2008) Lite hästhandel att köpa affärssystem, September 29: 18–19 (in Swedish).
9. Computer Sweden (2008) Enkelhet från svensk doldis, November 7(1): 6–7 (in Swedish).
10. Daghfous, A. (2007) Absorptive Capacity and Innovative Enterprise Systems: A Two-Level Framework, *International Journal of Innovation and Learning*, 4(1): 60–73.
11. Duchessi, P., Schaninger, C. M., and Hobbs, D. R. (1989) Implementing a Manufacturing Planning and Control Information System, *California Management Review*, 31(3): 75–90.
12. Engwall, M. (1995) *Jakten på det effektiva projektet*, Dissertation, Nerenius & Santérus, Uppsala (in Swedish).
13. Feeny, D., and Willcocks, L. (1998) Core IS Capabilities for Exploiting IT, *Sloan Management Review*, 39(3): 1–26.
14. Fredriksson, O., and Arola, M. (2009) En fallstudie av ett framgångsrikt affärssystembyte. In Hedman, J., Nilsson, F., and Westelius, A. (eds.), *Temperaturen på affärssystem i Sverige*, Studentlitteratur, Lund, pp. 167–196 (in Swedish).
15. Goldsmith, R. F. (2007) *REAL CHAOS, Two Wrongs May Make a Right*. Retrieved March 30, 2009 from: <http://www.compaid.com/caiinternet/ezone/goldsmith-chaos.pdf>.
16. Gulliksen, J., Göransson, B., Boivie, I., Blomkvist, S., Persson, J., and Cajander, Å. (2003) Key Principles for User-Centred Systems Design, *Behaviour & Information Technology*, 22 (6): 397–409.
17. Keil, M., Cule, P. E., Lyytinen, K. A., and Schmidt, R. C. (1998) A Framework for Identifying Software Project Risks, *Communications of the ACM*, 41(11): 76–83.
18. Lee, Z., and Lee, J. (2000) An ERP Implementation Case Study from a Knowledge Transfer Perspective, *Journal of Information Technology*, 15(4): 281–288.
19. Ljung, L. (2003) *Utveckling av en Projektivitetsmodell*, Licentiate Thesis, Linköping University (in Swedish).

20. Mathiassen, L., and Pries-Heje, J. (2006) Editorial: Business Agility and Diffusion of Information Technology, *European Journal of Information Systems*, 15(2): 116–119.
21. Mohr, L. B. (1982) *Explaining Organizational Behavior. The Limits and Possibilities of Theory and Research*. Jossey-Bass Publishers, San Francisco, CA.
22. Nerur, S., Mahapatra, R., and Mangalaraj, G. (2005) Challenges of Migrating to Agile Methodologies, *Communications of the ACM*, 48(5): 73–78.
23. Nilsson, A. (2008) *Projekttledning i Praktiken*, Doctoral dissertation, Umeå School of Business (in Swedish).
24. Rockart, J. F. (1979) Chief Executives Define Their Own Data Needs, *Harvard Business Review*, 57(2): 81–93.
25. Scott, J. E., and Vessey, I. (2000) Implementing Enterprise Resource Planning Systems: The Role of Learning from Failure, *Information Systems Frontiers*, 2(2): 213–232.
26. Shanks, G., and Seddon, P. (2000) Editorial, *Journal of Information Technology*, 15(4): 243–244.
27. Sia, S. K., and Soh, C. (2007) An Assessment of Package–Organisation Misalignment: Institutional and Ontological Structures, *European Journal of Information Systems*, 16(5): 568–583.
28. Soh, C., and Markus, M. L. (1995) How IT Creates Business Value: A Process Theory Synthesis. In DeGross, J. I., Ariav, G., Beath, C., Høyer, R., and Kemerer, C. (eds.) *Proceedings of the Sixteenth International Conference on Information Systems*, ACM Publications, New York, pp. 29–41.
29. Soh, C., and Sia, S. K. (2004) An Institutional Perspective on Sources of ERP Package–Organisation Misalignments, *Journal of Strategic Information Systems*, 13(4): 375–397.
30. Sumner, M. (2005) *Enterprise Resource Planning*. Pearson Education, Upper Saddle River, NJ.
31. The Standish Group International (2001) *Extreme CHAOS*. Retrieved March 30, 2009 from: <http://www.smallfootprint.com/Portals/0/StandishGroupExtremeChaos2001.pdf>.
32. Weill, P. (1992) The Relationship Between Investment in Information Technology and Firm Performance: A Study of the Valve Manufacturing Sector, *Information Systems Research*, 3(4): 307–333.
33. Willcocks, L. P., and Sykes, R. (2000) The Role of the CIO and IT Function in ERP, *Communications of the ACM*, 43(4): 32–38.

Patterns-Based IS Change Management in SMEs

Janis Makna and Marite Kirikova

Abstract The majority of information systems change management guidelines and standards are either too abstract or too bureaucratic to be easily applicable in small enterprises. This chapter proposes the approach, the method, and the prototype that are designed especially for information systems change management in small and medium enterprises. The approach is based on proven patterns of changes in the set of information systems elements. The set of elements was obtained by theoretical analysis of information systems and business process definitions and enterprise architectures. The patterns were evolved from a number of information systems theories and tested in 48 information systems change management projects. The prototype presents and helps to handle three basic change patterns, which help to anticipate the overall scope of changes related to particular elementary changes in an enterprise information system. The use of prototype requires just basic knowledge in organizational business process and information management.

Keywords Information system · Change management · Change patterns

1 Introduction

Globalization and turbulence of external environment require from different types of enterprises including SMEs the ability to change fast. One of the essential problems of enterprises in change management is the possibility to align business-driven changes to corresponding changes in enterprise information systems (ISs). While large organizations are equipped with highly professional IS staff, change management procedures, and standards, SMEs usually have none of these change supporting entities [1–3]. Besides, change management literature usually gives only

J. Makna (✉)

Institute of Applied Computer Systems, Riga Technical University, Riga, Latvia
e-mail: promis@apollo.lv

general guidelines how to manage change requests [4] or how to deal with, e.g., resistance to change [5, 6], but does not provide any practical help for IS change management in SMEs in particular. As a result, most of IS change management guidelines and standards are either too abstract or too bureaucratic to be easily applicable in small enterprises.

In this chapter research work on IS change management support in SMEs is presented. The aim of the research was to develop means for IS change management that would be applicable in SMEs, i.e., they would help to manage IS change without the involvement of expensive consulting companies and would be understandable and manageable by the SMEs staff, which has only basic knowledge in business process and information management. To achieve this aim basic change elements that are relevant in IS change situations were theoretically revealed and their change options were identified. Then the change options were analyzed against several IS theories in order to obtain change patterns of change options. The obtained change patterns were validated for 48 IS change cases. Three most common change patterns were built in the change management tool prototype. The purpose of the tool is to help business and information managers of SMEs to decide upon, anticipate, and control IS changes in their enterprises. The prototype of the tool demonstrates that the tool is easy to manage and helps to understand the scope of changes when at least one change option is known or intended.

The chapter is structured as follows. Section 2 briefly discusses research approach and the way how basic change elements were revealed and theoretically verified. Section 3 presents change options of change elements and shows how change patterns were obtained and analyzed as well as how most common change patterns were identified. Section 4 discusses the method and results of change pattern validation. The prototype of IS change management tool is presented in Section 5. Section 6 consists of brief conclusions.

2 Identifying Basic Change Elements

Analysis of change management literature revealed that systemic approach to change management is possible only if the correspondence between different change options can be controlled [7, 8]. However, the descriptions of change options available in IS change management literature were too abstract to be useful for designing a tool for IS change management in SMEs. Therefore the following research actions were taken:

- elements, which are eligible for change representation, were identified;
- change options of these elements were identified; and
- patterns of simultaneous occurrence of change options were analyzed.

In this section we discuss how change elements were identified and verified. In order to identify the change elements (1) IS definitions were analyzed, (2) business

process (BP) definitions were analyzed taking into consideration that the main task of the IS is to support business processes [9], (3) the results of IS and BP analyses were integrated to obtain the final set of IS change elements, and (4) the obtained set of IS change elements was validated using several enterprise architecture frameworks.

Based on the analysis of IS definitions [7, 10] the following elements characterising IS were identified: (1) data and information which is required by IS users, (2) IS users who use data and information to achieve their goals, and (3) an automated or manual collection of people or machines that process the data. Consequently, the following IS change elements characterizing IS were established: *IS activities*, *Data*, and *IS users*.

The elements which characterize the BP were obtained by analyzing BP definitions. Depending on different BP aspects there are the following BP definitions types: (1) BP definitions based on the description of transformation, (2) BP definitions based on the process description, and (3) BP definitions based on the interconnection description. Common elements of BP definitions are the following: (1) transformation or activities, (2) people who fulfill activities, (3) knowledge which is necessary to carry out the activities, (4) territory where activities are performed, (5) resources which are needed for the fulfillment of the activities, (6) control of collaboration of persons which are involved in BP, and (7) product or service which is produced during the activities. The final set of change elements was obtained unifying subsets of elements characterizing IS and BP. It consists of *Data*, *Knowledge*, *IS users*, *IS activities*, *BP activities*, *Control*, *Territory*, *Resources*, and *Product*.

The obtained set of elements was analyzed to ensure the completeness of the set and relevance of the elements. The analysis was performed by evaluating the role of the change elements in different enterprise architectures. The architectures chosen for validation are represented in the columns of Table 1. Different views offered by the architectures were crosschecked with the change elements. The abbreviations of the views, corresponding to particular change elements, are represented in cells of Table 1. The architecture TOGAF [11] offers the following views: information view (TO:INF), application view (TO:APP), technology view (TO:TEHN), and business view (TO:BUS). Architecture RM-ODP [12] offers the following viewpoints in organization architecture: enterprise viewpoint (RM:BIZ), information viewpoint (RM:INF), computational viewpoint (RM:FUNC), engineering viewpoint (RM:COM), and technology viewpoint (RM:TEHN). Zachman architecture [13, 14] offers the following views: data view (Z:DATA), function view (Z:FUNC), place view (Z:WHERE), person view (Z:WHO), time when function performs view (Z:WHEN), and motivation view (Z:WHY). DOD architecture [15] proposes the following views: (1) organization view (type of organization, occupation specialties) (DOD:O), (2) operational activities and tasks (DOD:A), (3) information elements (DOD:D), (4) systems view (facilities, platforms, units, and locations) (DOD:T), (5) IS applications (DOD:IS1), (6) IS activities (DOD:IS2), (7) triggers and events view (DOD:Z),

Table 1 Example table

Element	TOGAF	RM-ODP	Zachman	DOD	GERAM	Cimosa
Data	TO:INF	RM:INF	Z:DATA	DOD:D, DOD:S	GER:A	CIM:I
Knowledge	TO:APP	RM:IZS	Z:WHY	DOD:Z, DOD:R	GER:A	CIM:O
IS users	TO:INF	RM:FUNC	Z:WHO	DOD:S	GER:A	CIM:O
IS activities	TO:APP TO:TEHN	RM:TEHN	Z: FUNC	DOD:IS1, DOD:IS2, DOD:S	GER:P, GET:T	CIM:F
BP activities	TO:BUS	RM:FUNC	Z: FUNC	DOD:A	GER:P, GER:T	CIM:F
Control	?TO:BUS	RM:COM	Z:WHEN	DOD:K	GER:P	CIM:F
Territory	?TO:BUS	?RM:BIZ	Z:WHERE	DOD:O, DOD:T	?	CIM:R
Resource	?TO:BUS	?RM:BIZ	?Z: FUNC	DOD:IS1	?	CIM:R
Product	?TO:BUS	?RM:BIZ	?Z: FUNC	?	?	CIM:R

(8) performance parameters (DOD:K), (9) technical standards (data processing, transfer, security) (DOD:S), and (10) technology view (systems and standards) (DOD:R). GERAM consists of several components [16]. The GERA (Generalized Enterprise Reference Architecture) component corresponds to this research. GERA proposes the following three concepts: human-oriented concepts (GER:A), process-oriented concepts (GER:P), and technology-oriented concepts (GER:T). CIMOSA (CIM open system architecture) [17] surveys the organization from four perspectives: functional (CIM:F), information (CIM:I), resource (CIM:R), and organization (CIM:O).

To test the completeness of the set of change elements it is necessary to verify whether all architectural views include at least one change element. In Table 1 all views of architectures correspond to at least one change element. So, all changes in architectures views can be perceived by the change elements. Thus it is possible to consider that the set of elements is complete.

To identify the relevance of the elements, the level of abstraction at which the element appears in the architectural view was analyzed. Elements *Data*, *Knowledge*, *IS users*, *IS activities*, and *BP activities* are depicted with separate views of architectures. It means that these elements directly correspond to views of architectures. Elements *Control*, *Territory*, *Resource*, and *Product* are depicted only in components of the views or at the lower level of detail or abstraction of the views. In Table 1 these cells are depicted by a question mark. This indicates that these elements have less relevance in a particular enterprise architecture.

Considering the improvement cases of enterprises [18] changes occur in organization products or services, resource, and control. Therefore, to identify the changes of elements explicitly, the elements *Control*, *Territory*, *Resource*, and *Product* are included in the set of basic change elements.

3 Change Options and Change Patterns

In order to identify how change elements may alter during IS or BP changes in organizations, it is necessary to define possible change options for each element. To identify all change options of all identified change elements, more than 60 theories [19] and methods of IS and BP change management and reengineering were analyzed. Established change options of elements are presented in Fig. 1.

Change options of elements and relation between them were identified by answering two questions: (1) what changes take place in each element and (2) what elements must be changed according to a particular theory or method. Four change options for element *Knowledge* were identified: *new*, *improved*, *handed over*, and *received*. Due to the fact that there are no commonly accepted methods for how to measure improvement of knowledge, the change options *improved* and *new* of element *Knowledge* were united. In patterns these change options appear together and

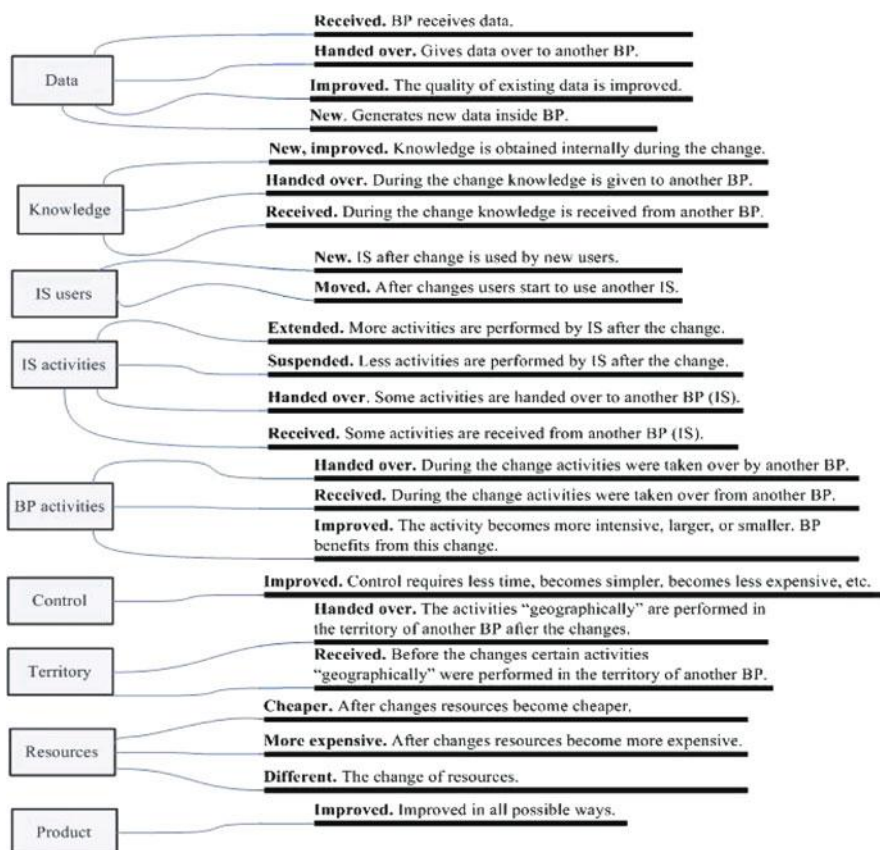


Fig. 1 Change elements (in boxes) and change options (in bold). All elements have change option "no change," which is not shown here

Table 2 Specific patterns of element changes established from the theories

Data	IS Knowledge users	IS activities	BP activities	Control	Territory	Resources	Product
1. Theory of administrative behavior [20], organizational knowledge creation theory [21]	New, Impr		Impr				
2. Language action theory [22], transactive memory theory [23], knowledge-based theory of the firm [24]							
Handed over	Rec		Rec				
3. Language action theory [22], transactive memory theory [23], knowledge-based theory of the firm [24]							
Rec	Handed over		Handed over				
4. Theories describing relation between activities of employees and data. Media richness theory [25], argumentation theory [26], Toulmin’s layout of argumentation [27], cognitive fit theory [28]							
Impr, New			Impr				
5. Transaction cost theory [29]							
Impr, New		Impr	Impr	Impr		Impr	Impr

their integration does not change the structure of the patterns. During the analyses of theories and methods 14 specific patterns of element changes were identified. Five of them are exemplified in Table 2. Each two rows in Table 2 shows a specific change pattern, where the source theories are given in the first row and the pattern is represented in the second row. The columns of the table show the change option of each element named in the title of the corresponding column.

Considering 14 specific patterns it was found that only the element *BP activities* has change options in all patterns. Therefore, all specific patterns were grouped according to the change options of this element. As a result, three basic patterns were obtained. The first pattern, further in the text called “Internal,” depicts changes in BP when BP internal possibilities are used. The second pattern, further in the text called “Extrec,” depicts changes of elements when BP receives activities from related BP or external environment. The third pattern, further in the text called “Extsend,” depicts changes of elements when BP sends (hands over) some activities to related BP(s).

4 Validity of Basic Change Patterns

The research hypothesis was that three basic change patterns introduced in the previous section are the dominant ones in IS and BP change situations. This hypothesis was tested in 48 real-life IS change management projects in SMEs. The duration of these projects varied from 1 month to 5 years. The SMEs were private, public, and government institutions involved in different types of business in different regions of Latvia: trade companies, financial institutions, telecommunication, and transportation enterprises. During the test, the states of all elements were registered before the change projects started and after the change projects ended using change

Table 3 The results of analyses of 48 IS change projects

	Data new	Data no change	Data hand over	Data received	Data improved	Total of rows
BP action. No change	4				1	5
BP action. Hand over	1	2	2	4	1	10
BP action. Received	3		4	3		10
BP action. Improved	16	2		3	2	23
Total of columns	24	4	6	10	4	48

elements and change options reflected in Fig. 1. Detected change values of all elements were organized in the table, where 48 rows show the particular projects and 32 columns show change options of all elements. There were 23 options from Fig. 1 and 9 options “no change” of each element. To identify the relations between elements a table with 32 rows and 32 columns was created. Rows and columns in this table reflected change options of the change elements. For each change option, related changes in all IS projects were counted and the sum of them represented in the cells of corresponding columns. An example of part of the table is shown in Table 3 (totals illustratively are given for the exemplified part of the table only).

To characterize the strength of relationships between elements, the category data analysis method [30] was used. Category data analysis method allows characterizing numerically the relationships between several elements. Numerical characterization of the relationship allows distinguishing between strong and weak relationships. According to category data analyses method [30] two variables of a matrix are independent if the value of matrix row i and column j is equivalent to $n_{ia} \times n_{aj}/n$, where n is the number of experiment, n_{ia} the total of row i , and n_{ja} the total of column j . Thus deviation from independence in this cell can be expressed as

$$D_{ij} = n_{ij} - n_{ia} \times n_{aj}/n \quad (1)$$

where n_{ij} is the number of experiments in cell when two elements have change options represented by row i and column j .

The formula (1) was applied to all elements of Table 3. As a result, the table with constants was obtained. Part of the resulting table is presented in Table 4. Table 4

Table 4 Deviation from dependence of *BP activities* and *Data elements*

	Data new	Data no change	Data hand over	Data received	Data improved
BP action. No change	1.5	−0.416	−0.625	−1.04	0.58
BP action. Hand over	−4.0	1.16	0.75	1.91	0.17
BP action. Received	−2.0	−0.83	2.75	0.916	−0.83
BP action. Improved	4.5	0.084	−2.87	−1.79	0.08

Table 5 Deviation from dependence of *BP activities* and *Data* elements expressed by ratios in interval between 0 and 1

	Data new	Data no change	Data hand over	Data received	Data improved
BP action. No change	0.647	0.4216	0.397	0.3482	0.5388
BP action. Hand over	0.00	0.6070	0.5588	0.6952	0.4905
BP action. Received	0.2352	0.3729	0.7941	0.5783	0.3729
BP action. Improved	1.00	0.4804	0.1329	0.2599	0.4799

shows the deviation from dependence expressed by ratios. The sum of each row and column is equivalent to zero. The range of ratio values in the table is from -4 to 4.5 . To move the ratio interval to interval from 0 to 1 for each ratio constant value 4 was added and the sum multiplied by 0.11764. Ratio 0.11764 is obtained by dividing 1 to the new highest value 8.5. The results are presented in Table 5.

The ratio in the cell characterizes the deviation of dependence between the change options of two elements. Higher values of ratios specify a stronger relationship between the change options. It means that if one element has this change option, then the other element changes to achieve a corresponding change option. The analysis of results, presented in Table 5, for change options of all change elements shows that patterns “Internal,” “Extsend,” and “Extrec” described in the previous section are dominant. It means that during changes in SMEs the change elements will most commonly have change options that correspond to these patterns. The patterns are described in greater detail in the next section.

5 The Prototype for IS Change Management Support in SMEs

The prototype of the developed information systems change management tool maintains three basis change management patterns that were obtained by classification of 14 patterns according to change option of change element *BP activities* (Sections 3 and 4). Each pattern represents a particular set of change options, which complement one another during one change management cycle and thus may be considered as occurring “simultaneously,” i.e., complementing one another. The prototype provides graphical representation of each pattern (see fragments of the prototype in Figs. 2, 3, and 4 and overview of the prototype in Fig. 5).

Pattern “Internal” (Fig. 3) represents IS changes occurring in situations when changes in BP are accomplished using internal resources only. This means that there



Fig. 2 Change palette of Pattern “Internal”

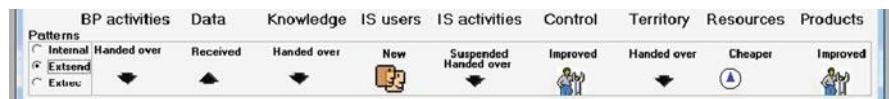


Fig. 3 Change palette of pattern “Extsend”



Fig. 4 Change palette of pattern “Extrec”

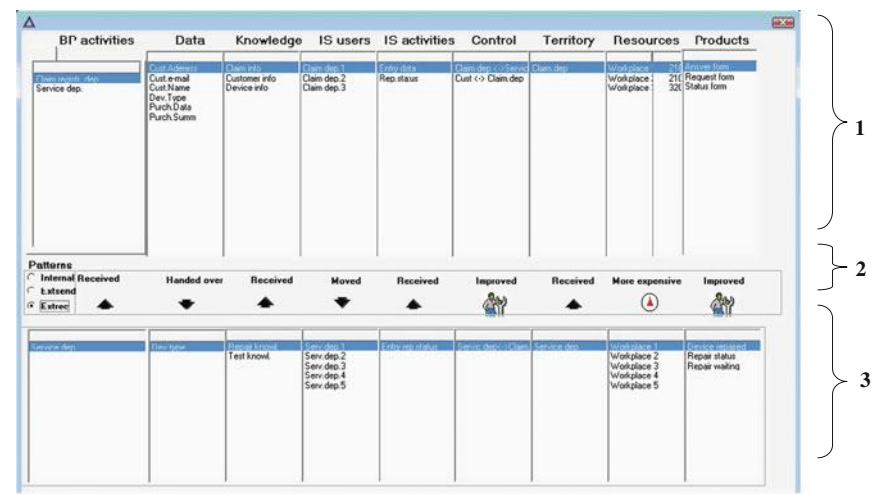


Fig. 5 The main window of the prototype of IS change management tool (1 – the process in interest, 2 – change palette (3 options), 3 – other process(es))

are no changes in IS change elements *IS users* and *Territory*. The pattern shows that in this case most commonly new data not used before will be identified or the quality of data improved. Due to new/improved data new knowledge is available. In order to support the new data and knowledge, the use of IS activities is extended. As a result of their improvement, BP activities changes lead to improved BP control, products, and use of BP resources. By considering the pattern reflected in Fig. 3, BP and information managers may see that all presented changes are likely to accrue if they introduce one of them. The upper window of the prototype screen (Fig. 5) is used for recording particular changes expected with respect to each change option presented by the pattern.

Pattern “Extsend” occurs in situations when part of BP activities is handed over to another BP(s) (Fig. 3). In this case the BP receives the data of the results of activities, which were handed over to other processes. The territory of certain BP activities changes from one BP to another BP. In order to support the fulfillment

of activities the BP also hands over knowledge about the activities. From IS point of view, the IS receives data, gets new IS users, and suspends or hands over some part of IS activities. These changes improve BP control, resource utilization, and products. The business process of interest is represented in the upper window, other processes in the lower window of the prototype (Fig. 5). The users of the prototype may enter and move particular instances of change elements from one window to another until the change pattern is fully represented by change element instances.

Pattern “Extrec” occurs in situations when BP activities are received from another BP (Fig. 4). This case is similar to pattern “Extsend.” Difference is that BP receives new activities instead of handing them over, i.e., the BP receives activities and knowledge from another BP and hands over the data about fulfillment of received activities.

The use of patterns contributes to IS change management in SMEs as follows. (1) The patterns help to identify complementary changes to changes that have already occurred. (2) The patterns allow ensuring that the full scope of organizational changes is anticipated. (3) The tool (Fig. 5) enables to experiment with different instances of change elements according to pattern prescribed change options.

The main window of the change management tool’s prototype consists of three basic parts (Fig. 5). Information about the process to be analyzed is entered in Part 1. The change elements of BP are described here by corresponding values/parameters/instances. Part 2 gives an opportunity to choose the change pattern. After it is chosen a corresponding change palette appears (Figs. 2, 3, and 4). In each palette every change element is visually supported by an appropriate icon. If patterns “Extsend” or “Extrec” are chosen, Part 3 of the window may be utilized. In this part information about all processes except the one that is analyzed may be entered.

The tool can be used for the following main purposes: (1) to ensure completeness of change management; (2) to develop new IS or BP; and (3) to investigate what are the complementary changes of particular changes in BP (or IS) elements. For instance, if we want to know how IS or BP could be improved, the tool could be used as follows: (1) Title of BP is entered in Part 1 of the window (right-hand sub-window). Change element values/instances/parameters are entered in the neighboring sub-windows. (2) In Part 2 of the window pattern “Internal” is chosen (3). In Part 1 possible changes of each parameter are considered according to suggestions reflected in the palette of the pattern. If necessary it is possible to add new or more detailed parameters. It is necessary to analyze whether the changes that are possible will improve the business process. (4) The analysis similar to that described in the previous point is accomplished using patterns “Extsend” and “Extrec.” That is how several alternatives for BP improvement may be obtained.

6 Conclusions

The approach, the method, and the prototype for IS change management support in SMEs were presented in this chapter. The strengths of the proposed solution are in the utilization of deep and thorough IS knowledge in the form of simple IS change

management tool. The use of the tool would enable IS change managers to think about IS change management systemically and pragmatically that can lead to professionally sound, timely, and implementable IS change requests and supplementary organizational support in SMEs.

Currently only basic change management patterns are shown in the prototype of the IS change management tool. These patterns in Table 5 (Section 4) were represented by the highest ratios depicting the dominating role of corresponding changes. Nevertheless non-dominating relationships exist, too. This means that basic patterns do not dominate in all cases. The aim of future research is to investigate what are the causes of those relationships, which are not included in the basic change patterns. This would help to enrich three basic patterns with conditional extensions for deeper analysis of change situations.

References

1. Robinson, P., Gout, FL.: Extreme Architecture Framework: A minimalist framework for modern times. In: Saha, P. (ed.) *Handbook of Enterprise Systems Architecture in Practice*, IGI Global, pp. 18–36 (2007).
2. Goikotxea, A.: *Enterprise Architecture and Digital Administration: Planning Design and Assessment*. World Scientific Publishing Co. Pte. Ltd., Singapore (2007).
3. Harrington, H.J., Esselding, E.C., Nimwegen H.: *Business Process Improvement. Workbook. Documentation, Analysis, Design and Management of Business Process Improvement*. McGraw-Hill (1997).
4. Daoudi, F., Nurcan, S.: A Benchmarking Framework for Methods to Design Flexible Business Processes. In: *Software Process Improvement and Practice*, pp. 51–63 (2007).
5. Mumford, E.: *Redesign Human Systems*. Information Science Publishing, UK (2003).
6. Zacarias, M., Caetano, A., Magalhaes, R., Pinto, H.S., Tribolet, J.: Adding a human perspective to enterprise architectures. In: *Proceedings of 18th International Workshop on Database and Expert Systems Applications*, pp. 840–844 (2007).
7. Maddison, R., Dantron, G.: *Information Systems in Organizations. Improving Business Processes*. Chapman & Hall, London (1996).
8. Skalle, H., Ramachandran, S., Schuster, M., Szaloky, V., Antoun, S.: *Aligning Business Process Management, Service-Oriented Architecture, and Lean Six Sigma for Real Business Results*. IBM Redbooks (2009).
9. Spadoni, M., Abdomoleh, A.: Information Systems Architecture for business process modeling. In: Saha, P. (ed.) *Handbook of Enterprise Systems Architecture in Practice*, IGI Global, pp.366–380 (2007).
10. Alter, S.: Defining information systems as work systems: implications for the IS field. *European Journal of Information Systems* (2008) 17. Operational Research Society Ltd. pp. 448–469 (2008).
11. <http://www.ibm.com/developerworks/library/ar-togaf1/#N10096>.
12. Reference Model of Open Distributed Processing, <http://en.wikipedia.org/wiki/RM-ODP>.
13. Zachman, J.: A Framework for Information Systems Architecture. *IBM Systems Journal*, 26(3) (1987).
14. Extending the RUP with Zachman Framework, <http://www.enterpriseunifiedprocess.com/essays/zachmanFramework.html>.
15. DoD Architecture Framework. Version 1.5 Volume 2. Product Description. http://www.defenselink.mil/cio-nii/docs/DoDAF_Volume_II.pdf.

16. GERAM: Generalized Reference Architecture Enterprise and Methodology. Version 1.6.3. IFIP – IFAC Task Force on Architectures for Enterprise Integration. <http://www.cit.gu.edu.au/~bernus/taskforce/geram/versions/geram1-6-3/v1.6.3.html>.
17. Nazzal, D.: Reference Architecture for Enterprise Integration. CIMOSA GRAI/GIM PERA <http://www2.isye.gatech.edu/~lfm/8851/EIRA.ppt#264,8,CIMOSAEnterprise>.
18. Teng J.T., Grover V., Fiedler K.D.: Initiating and Implementing Business Process Change: Lessons Learned from Ten Years of Inquiry. In: T. Grover V., Kettinger W. (eds.): Process Think: Winning Perspectives For Business Change In The Information Age. Idea Group Publishing, UK, pp.73–114 (2000).
19. Theories Used in IS Research, available at www.fsc.yorku.ca/york/istheory/wiki/index.php/Main_Page.
20. Theory of administrative behavior, http://www.fsc.yorku.ca/york/istheory/wiki/index.php/Administrative_behavior%2C_theory_of.
21. Organizational Knowledge creation theory, available at http://www.fsc.yorku.ca/york/istheory/wiki/index.php/Organizational_knowledge_creation.
22. Language action perspective, www.fsc.yorku.ca/york/istheory/wiki/index.php/Language_action_perspective.
23. Transactive memory theory www.fsc.yorku.ca/york/istheory/wiki/index.php/Transactive_memory_theory.
24. Knowledge-based theory of the firm http://www.fsc.yorku.ca/york/istheory/wiki/index.php/Knowledge-based_theory_of_the_firm.
25. Media richness theory, www.fsc.yorku.ca/york/istheory/wiki/index.php/Media_richness_theory.
26. Argumentation theory, www.fsc.yorku.ca/york/istheory/wiki/index.php/Argumentation_theory.
27. A Description of Toulmin's Layout of Argumentation <http://www.unl.edu/speech/comm109/Toulmin/layout.htm>.
28. Cognitive fit theory, www.fsc.yorku.ca/york/istheory/wiki/index.php/Cognitive_fit_theory.
29. Transaction cost economics, www.fsc.yorku.ca/york/istheory/wiki/index.php/Transaction_cost_economics.
30. Kendall, M. G., Stuart A.: the Advanced Theory of Statistics. Interference and Relationship. Volume 2. Charles Griffin & Company Limited, London.

Applying Use Cases to Describe the Role of Standards in e-Health Information Systems

Emma Chávez, Gavin Finnie, and Padmanabhan Krishnan

Abstract Individual health records (IHRs) contain a person's lifetime records of their key health history and care within a health system (National E-Health Transition Authority, Retrieved Jan 12, 2009 from <http://www.nehta.gov.au/coordinated-care/whats-in-iehr>, 2004). This information can be processed and stored in different ways. The record should be available electronically to authorized health care providers and the individual anywhere, anytime, to support high-quality care. Many organizations provide a diversity of solutions for e-health and its services. Standards play an important role to enable these organizations to support information interchange and improve efficiency of health care delivery. However, there are numerous standards to choose from and not all of them are accessible to the software developer. This chapter proposes a framework to describe the e-health standards that can be used by software engineers to implement e-health information systems.

Keywords Information systems · e-Health · Information exchange · Health care delivery

1 Introduction

There are a large range of e-health computer application categories. These vary from computer-based medical records that facilitate access to low-cost therapies, computer-based population, or community health records that are usually used in public health to trace different types of health hazards, telemedicine applications, and patient applications like health-oriented web sites.

E. Chávez (✉)

School of Information Technology, Bond University, Gold Coast, QLD, Australia
e-mail: emchavez@bond.edu.au

Standards set specifications, formats, terminologies, and others to enable information exchange. There are standards which have been developed for the same purpose offering two or more solutions [2]. Nevertheless, none of them may be universally acceptable. On the other hand, multiple standards are also important as this leads to competition and helps to promote the quality of the e-health system environment [3, 4]. As a result, all actors (e.g., developers, vendors, acquirers) in the e-health sector find it difficult to select the best or most relevant standard.

The RIDE project in the document “Requirements Analysis for the RIDE Roadmap” has identified a number of significant e-health application scenarios [5]. These scenarios are grouped into five categories:

1. management and exchange of clinical data,
2. applications for the patient,
3. telemedicine applications,
4. public health applications, and
5. other general applications.

The document also describes an overview of the main functionalities, actors, and interaction of e-health information.

The Centre for Health Informatics at Bond University in Australia has been working during the last 2 years on mapping the e-health standards infrastructure as a knowledge base to provide quality in the development process of e-health information systems. An extensive number of mature and popular standards are currently being used to support e-health areas such as interoperability and security. Nevertheless, some of the standards support only parts of the process while others support a mixture of different areas of the e-health arena.

In past research a taxonomy of e-health standards was developed to provide guidance to identify standards information in the specific IT domain of software engineering [6]. One-hundred and ten different e-health standards were reviewed to be used in different phases of the development of e-health information systems. This essentially proposed a classification scheme for standards. The scheme could be used as a guide to identify standards relevant to a particular area of software engineering.

As the scheme was designed only to identify standards, it did not describe the details of how a standard could actually be used. In this chapter we show how various standards can be used. We describe use cases to capture relevant functionalities of the e-health information system. These use cases are then extended with a selection of standards. Specifically, use case diagrams are used to identify and partition the system functionalities (scenarios) and thus define the scope of the functionalities being covered. This scope can then be related to the scope of the standards. We are using use cases to provide guidance in standard selection during analysis and design rather than as a documentation technique for designed or implemented systems. Thus, these use cases are more specific than essential use cases but not as specific as sequence diagrams.

In Section 2, a brief background and evolution of e-health standardization is shown. Section 3 introduces the framework of e-health actors and use case diagrams for some e-health applications are explained. We conclude the chapter in Section 4 with the conclusions and the future research possibilities.

2 e-Health Standardization

There are at least six principal organizations which have developed e-health international standards including ASTM-E31, ANSI-HL7, CEN-TC 251, ISO-TC 215, NEMA-DICOM, and IEEE with the family of 1073.X standards.

ASTM, the American Society for Testing and Materials based in the United States, is mainly used by commercial laboratory vendors. Its committee E31 is focused in developing e-health standards.

ANSI, the American National Standards Institute, operating in the United States, is developing HL7 which is a family of standards for the exchange, integration, sharing, and retrieval of electronic health information.

CEN, the European Committee for Standardization, has formed the Technical Committee CEN/TC 251 Health Informatics, which has created a series of European pre-standards and standards covering the electronic exchange of medical data principally focused on electronic health records.

ISO, the International Organization for Standardization, develops e-health standards through the technical committee ISO TC215, which involves a number of other organizations such as CEN and HL7. The American College of Radiology (ACR) and the National Electrical Manufacturers Association (NEMA) have published DICOM, a standard that addressed the methods for data transfer in digital medical images in the United States.

Finally IEEE, the Institute of Electrical and Electronics Engineers, is establishing a series of standards related to medical device communications.

Some of the standards that are relevant in the implementation of e-health applications are

- Standards that set communication protocols to enable systems integration and information exchange such as HL7, DICOM, and the ISO/TR 18307:2001 Interoperability and compatibility in messaging and communication standards.
- Standards that set terminologies and machine-readable terminologies such as SNOMED.
- Standards that provide plug and play interoperability for patient-connected medical devices such as IEEE P11073-00103 Point-of-care medical device communication.

Depending on the domain and scope of the application, appropriate standards need to be followed. In the next section we outline a few scenarios and how UML can be used.

3 e-Health Application Scenarios

3.1 *Using UML to Describe e-Health Functionalities and Standards Incidence*

A conceptual model is a picture describing a real-world domain which is independent of any aspects of information technology. Additionally, the information system design phase is concerned with describing the information system through design models intended to characterize the software system. UML can be considered as an effective tool to support both of these processes [7].

Although UML has nine types of diagrams, in this chapter we apply use case diagrams to model the functionality of a system using actors and use cases. In use cases, an actor is an outsider (cannot be controlled by the system) who interacts with the system. As a basis for our approach, we assume that developers specify the functionality of the e-health information system under consideration via use cases.

As there is no uniform way to describe the role of standards in behaviours, we present a number of examples to illustrate our approach. Initially we use the diagrammatic representation of uses cases to describe our annotations. In the final example we use the flow of events to show how the role of the standards can be specified more precisely.

3.1.1 Use Case 1: System Registration

Health care providers and patients are the major users of e-health applications. Each of them has specific roles and requirements for collecting, retrieving, processing, and displaying health care data. Legislative requirements also dictate who can have access to the data and what processing they can perform. Therefore access control is a key functionality to any e-health information system.

The actors involved are system user (patient and/or health care professional) and system administrator who is in charge of managing the user's rights and privileges.

Figure 1 shows the registration main functionalities to be considered in any e-health information system. To register a user the definition of roles and the generation of a unique patient identifier like the universal health care identifier (UHID/ID) are essential functions. This is supported by the relevant ASTM standard. The role definition and privilege management are supported either for the ASTM E1986 standard or for the ISO22600-3 standard. The three standards shown in Fig. 1 are used by two separate behaviours with the same use case.

3.1.2 Use Case 2: Diagnostic Imaging

Diagnostic imaging produces images and doctors can request them as complementary diagnostic information. In most cases the doctor is given a written report only.

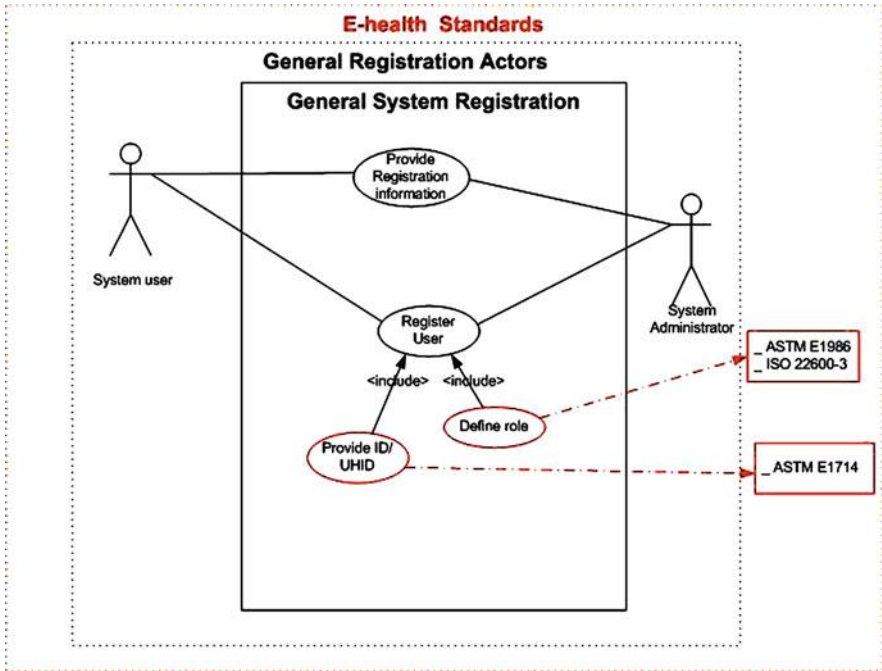


Fig. 1 Use case registration

The actors involved are the patient, medical requester (any practitioner who requests an image to be realized of the patient’s body or organ), and the medical imagist who is the person responsible for generating and handling the image after a suitable examination.

The DICOM standard enables storage and handling of the image. It also offers a messaging protocol (like HL7 and CR14300) to transfer data such as reports and the actual image between different information systems (Fig. 2). In the case of an electrocardiography (ECG) the behaviour could also be governed by the standard EN 1064:2005 that allows standardized transfer of digitized ECG data and results between various computer systems. It also specifies data format for ECG reports and data exchange from any vendor’s computerized ECG recorder to any other’s vendor central ECG management system [8].

3.1.3 Use Case 3: Laboratory Results

This use case involves the functionalities required to report laboratory test results. Only two actors are involved here: the health professional and the laboratory engineer.

Figure 3 shows that the activities of reporting and transmitting laboratory results can be supported by e-health standards. LOINC [9] provides a data set of universal

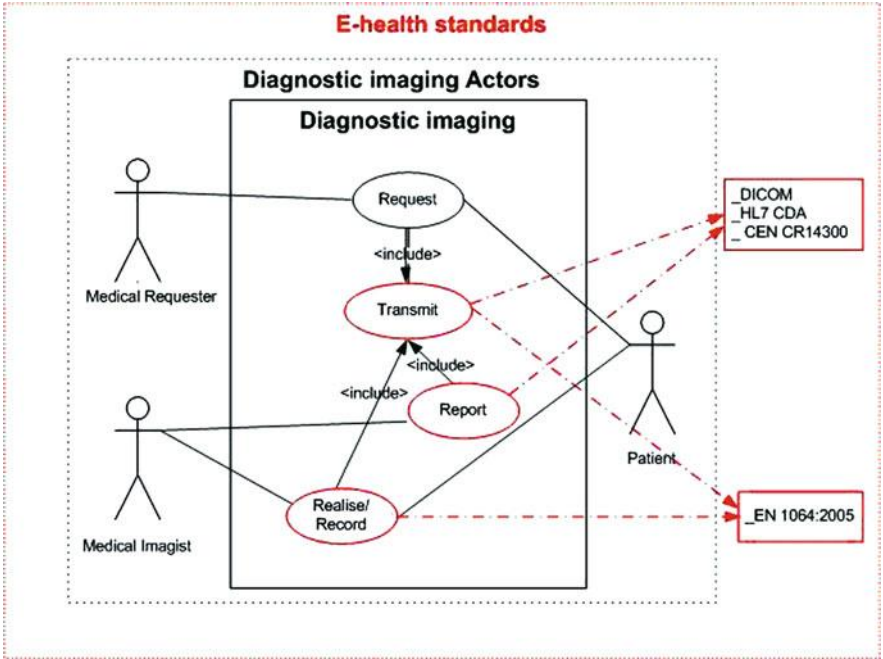


Fig. 2 Diagnostic imaging (adapted from Ride [5])

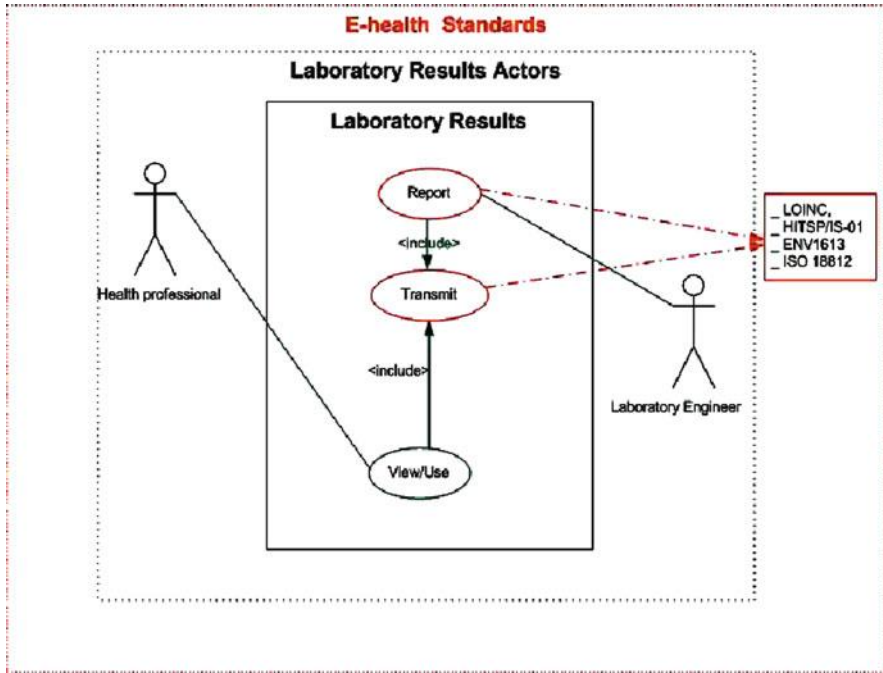


Fig. 3 Use case laboratory results

identifiers (universal codes and names) to identify laboratory and other clinical observations facilitating the exchange and pooling of clinical results. The standards HITSP [10], ENV 1613, and ISO 18812 specify messages for electronic information exchange between analytical instruments and laboratory information systems within a clinical laboratory.

3.1.4 Use Case 4: Electronic Prescribing

This use case describes the functionalities required to create and manage a prescription submitted by a health professional and transmitted to a pharmacist or an organization that dispenses the medication.

The actors involved are the patient, the prescriber, and the dispensing agent. It can be seen from Fig. 4 that there are four main standards that apply to e-prescription. The ENV 12610 standard sets the base of knowledge and terminologies to prescribe and dispense medicines, while the ENV 13607 [11] standard provides the guidelines to support information exchange within health care entities that prescribe, dispense, or administer medicinal products.

The Ride project document [5] describes the flow of information for each functionality of the use case. Using that as a basis, we have annotated the flow with the standards that apply to each activity. Section 3.2 gives some examples about the relevant standards that apply in the implementation of an e-prescribing system.

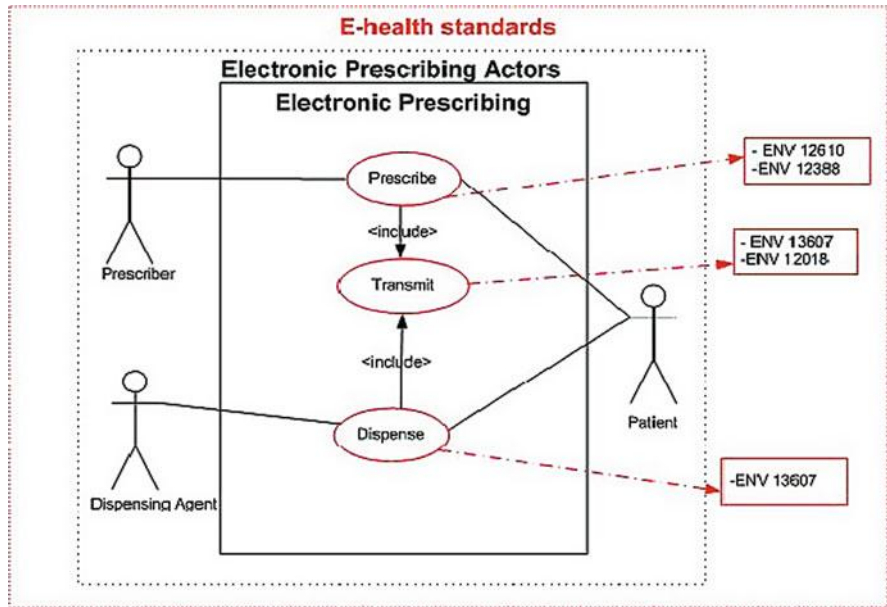


Fig. 4 Electronic prescribing (adapted from Ride [5])

3.2 Use Case Flow Description

A particular flow of events extended with appropriate standards is described below.

- Prescribe:
 - The prescriber prepares prescription for one or multiple drugs. Here the standard ENV 12610 is applicable. The standard specifies all information necessary to convey the amount of a medicinal product that has been taken or prescribed to be taken in a certain time interval. It includes measurements of dose units, number of dose units, relevant dosing, and reasons for taking the medication.
 - The prescriber signs the prescription. Here the standard ENV 12388 which sets the use of the RSA algorithm for all digital signatures in health care is applicable.
- Transmit:
 - Option 1 – Prescription is delivered to the e-health system: The standard ENV 13607 can be used here. This specifies a message, new prescription message, for electronic prescribing of medicinal products sent from the prescriber to a dispensing agent, possibly via a relaying agent.
 - Option 2 – Prescription is stored on a portable storage medium: The standard ENV 12018 applies here. This standard offers a common framework for structured data, used in temporary connecting devices to the system, such as electronic cards.
 - Option 3 – Prescription sends prescription to the pharmacy of choice: Here the standard ENV 13607 can be used. The relaying agent sent the prescription message to the selected dispensing agent.
- Dispense:
 - Pharmacist acknowledges receipt of prescription: The standard ENV13607 applies here. A prescription dispensing report is sent to the prescriber or to an alternate destination. An appropriate cautionary and advisory label from the pharmacist is also included in the message.

3.3 Summary

As can be seen with these examples, use cases are used to describe the functionalities and behaviour of a particular e-health information system. System functionalities and actors are described to understand the sequence of actions that the system must perform.

Although use cases have been designed mainly to capture requirements, we have added an external attribute to tie functionalities to implementation constraints. In

this case, giving advice of which standards apply supports the implementation of determined system functionalities.

By knowing the system functionalities a matching e-health standard (in the case of an existing one) to support the implementation of a particular functionality has been added to each diagram. Therefore, developers will have advice in the design and implementation phase by knowing in advance system functionalities and the necessary conditions that the system will need to have to support and comply with specific standards.

4 Conclusions and Future Work

There are a large number of e-health standards. Knowing which one to use is a hard task for everyone involved in the implementation of health information systems. Although vendors do not necessarily need to comply with all existing e-health standards, they are interested in specific standards which are mandated by regulatory authorities within their jurisdiction.

In general there are numerous standards that seem to support the same functionalities [6]. There are also many areas that do not have any relevant standard. For that reason, mapping standards to behaviours in requirement specifications can help the task of software development. This chapter has demonstrated how this can be done via use cases. By examining the standards involved in each behaviour issues such as scoping and costing can be addressed.

Although initial indications are encouraging, we aim to analyse the role of these annotations in the software engineering process. This would then provide the evidence on the costs and benefits of using these annotations as part of the requirements specification.

We also plan to include the use of the tool Use Case Maker [12] to keep a record of all the standards information that we have for each use case provided.

References

1. National E-Health Transition Authority. (2004). A vision for individual health records. e-Health challenges for standardization. Retrieved Jan 12, 2009 from <http://www.nehta.gov.au/coordinated-care/whats-in-iehr>.
2. K. Gunnar. (2003). E-Health challenges for standardization. In: Workshop on Standardization in E-health, Geneva.
3. National E-Health Transition Authority. (2007). Standards for e-health interoperability.
4. Cooper, J. G. and Pauley, K. A. (2006). Healthcare software assurance. In: Annual Symposium of the American Medical Informatics Association, pp. 166–170.
5. RIDED2.3.1: Requirements analysis for the RIDE roadmap. (2006). Retrieved Jan 12, 2009 from http://www.srdc.metu.edu.tr/webpage/projects/ride/deliverables/RIDE-D2.3.1-2006-10-13_final.pdf.
6. Chavez, E., Finnie, G., and Krishnan, P. (2008). A taxonomy of e-health standards to assist system developers. Proceedings of the 17th International Conference on Information Systems Development (ISD2008).

7. Ambler, S.W. (2004). Agile Model Driven Development with UML 2. Cambridge University Press.
8. Health informatics – Standard communication protocol – Computer-assisted electrocardiography. (2005). Retrieved Jan 12, 2009 from <http://www.saiglobal.com/PDFTemp/Previews/OSH/IS/EN/2005/I.S.EN1064-2005.pdf>.
9. Logical Observation Identifiers Names and Codes (LOINC®) Users' Guide. (2009). Retrieved March 13, 2009 from <http://loinc.org/downloads/files/LOINCManual.pdf>.
10. HIST V2.0. (2008). Retrieved March 13, 2009 from <http://publicaa.ansi.org/sites/apdl/IOLib/Forms/AllItems.aspx>.
11. ENV 13607. (2000). Health informatics – Messages for the exchange of information on medicine prescriptions. Retrieved March 13, 2009 from http://www.medis.or.jp/2_kaihatu/iso/iso_tc215_wg5/data/part7_f_en13607.pdf.
12. Use case maker. (2008). Retrieved March 13, 2009 from <http://sourceforge.net/projects/use-case-maker/>.

Discussion on Development Trend of Chinese Enterprises Information System

Xiao-hong Gan

Abstract Chinese enterprises have constructed their information systems one after another for reducing production cost, improving working efficiency, and better adapting to the development of the information age as well as the market environment. This chapter analyzes the status quo of Chinese enterprise information system, states the reasons for driving the development of enterprise information system, and discusses the development trend of Chinese enterprise information system from many angles.

Keywords Information system · Enterprise informationization · Reasons · Development

With the rapid development of information technology and continuous innovation of management ideas, enterprise management information system has been recognized as a kind of vigorous management and a good method for realizing complex enterprise targets. In recent years, many successful Chinese enterprises prove that the construction of enterprise information system has become the basis for the enterprise's survival, development, and independent innovation. However, we can establish adaptive informationization development strategy according to the changes of internal and external environment of the enterprise only when we have an overall and in-depth knowledge of enterprise information system and have fully understood the reasons for driving the development of enterprise information system and its development trend.

1 Status Quo of Enterprise Information System

Modern management information system (MIS) is an integrated human-machine system that collects, transmits, processes, stores, renews, and maintains information

X.-h. Gan (✉)

Information College, Jiangxi University of Finance and Economics, Nanchang 330013, China
e-mail: xhg1596@163.com

through computers and relevant equipment, and supports the enterprise's high-level decision making, medium-level control, and primary-level operation, for the purpose of competitive advantage of enterprise strategy and improvement of benefit and efficiency.

In the late 1990s, China started the R&D of management information system and many institutions developed the management information system software. First, financial software achieved great success, which brought the development of other industries and made an indelible contribution to the modernization of Chinese enterprise management. The application of Chinese enterprise information system has a history of more than 10 years, and now the system has almost been involved with all fields such as national defense, government departments, manufacture, aviation, finance, electric power, traffic, communication, marketplace, manufacturing industry, various kinds of trade companies, etc., all of which work with high efficiency and high quality under the support of computer management information system. As compared with early applications, enterprise system has changed greatly, and the enterprise has entered into a higher level in terms of product, management, marketing, and service. The enterprise's external environment is completely different from that in the past. What we face is the speedup of economic globalization, enlargement of competitive scope, and fiercer competition.

Today, the construction of enterprise informationization has obtained favorable development, and many enterprises are actively developing or preparing for constructing enterprise information system. But due to lack of overall planning and uniform standard, management content and data definition are nonuniform, low-level repeat development problem is serious, more and more isolated information islands are formed, integration between systems is difficult, there is no scale, and the advantage and potential of whole benefit are not brought into full play.

2 Reasons for Development of Enterprise Information System

Today, with the rapid development of science and technology, the development of information technology and management information system promotes the modernization of enterprise management, improves the enterprise's ability in global competition, and also brings new opportunity for the development of future enterprise information system. Throughout the development history of enterprise information system, we can see that enterprise information system develops unceasingly under the common action of the development of science and technology; the change of management thought, management method, and competitive environment; the reform of enterprise management ideas; and others.

2.1 Development of Science and Technology

Enterprise information system can always timely absorb and utilize the latest development achievements of computer hardware and software technology, while new

computer technology and network communication technology promote the reform of new management thought and management operation method. For example, the development of LAN technology hastens MRP II (manufacturing resource planning) and promotes the integration development of enterprise internal information system; similarly, Internet technology has laid the foundation for ERP (enterprise resource planning), SCM (supply chain management), etc. The development of enterprise information system leaps over the organization boundary and continuously extends to downstream and upstream supply chain. Currently, with the development of wireless network technology, wireless network has become a genuine network “existing everywhere.” Through wireless access service, enterprise information system can provide more convenient service for any supervisor, so that the supervisor can easily make a decision or communicate with customers wherever he/she is. The above-mentioned is a new target of constructing enterprise information system.

2.2 Development of Management Thought and Management Method

Society and science & technology are always developing, and new management modes and management methods adapting to the knowledge-based economy continuously come forth, including agile manufacturing (AM), virtual manufacturing (VM), lean production (LP), customized manufacturing (CM), customer relationship management (CRM), supplier relationship management (SRM), Large-scale customization (LC), advanced planning & scheduling (APS) based on theory of constraints, electronic commerce (EC), commercial intelligence (CI), enterprise performance management (EPM) based on balanced scorecard, etc. So, enterprise management information system must continuously increase these new thoughts and new methods for adapting to the enterprise’s management reform and development requirements.

2.3 Change of Competitive Environment

The information in management information system develops from only limited flow inside the enterprise in pursuit of scale benefit to simultaneous flow both inside and outside the enterprise in pursuit of regional economy and global economy. As a strong information processing tool, information system can obviously improve an organization’s coordination and resource integration capacity, learning capacity, and reform capacity, and can play an important role in supporting the enterprise’s core competitiveness. For example, ERP system serves for the integration of enterprise internal resource, SCM strengthens the enterprise’s coordination capacity with supply chain partner, and knowledge management system enhances the enterprise’s learning capacity. Meanwhile, the application of information system creates a flat organization structure and promotes the reengineering and improvement

of enterprise business process, which will be favorable for establishing flexible operation mechanism and enhancing the enterprise's adaptive capacity to the rapidly changed external environment.

2.4 Reform of Enterprise Management Ideas

With the integration of global economy and fiercer competition, the trend of product homogeneity is more and more obvious, and the difference in price and quality is not the main means for an enterprise to obtain profits. The enterprise realizes the importance of meeting customers' individual requirements and even exceeding customers' demands and expectations. The key to success is to focus on customers, listen to customers' appeals, and rapidly respond to the customers' changing expectations, which result in the combination of ERP system reasonably and efficiently utilizing enterprise resources, CRM meeting customers' demands, and SCM in pursuit of the optimization of supply chain.

3 Development Trend of Future Enterprise Information System

Continuous development and change of global economy environment, rapid development of the Internet and information technology, as well as changeable competitive environment have changed enterprise management modes, methods of disposing matters, and people's lifestyle. The enterprise's management thoughts and management methods are innovated unceasingly. The overall development trend of the Chinese enterprise information system is a combination of various management thoughts and management modes, system application networking and development platform standardization, business process supportable application system integration, and intelligent information system development.

3.1 Combination of Various Management Thoughts and Management Modes

Many large-scale enterprises in China have developed and introduced many information systems operated on different operation systems and heterogeneous networks. On the one hand, these information systems execute certain function respectively, which causes information organization lack of standardization, repeated data collection, and increasingly serious "isolated information islands" problem; on the other hand, the enterprise urgently needs forming the chain of survival for common development with customers or suppliers and realizing information share and direct data exchange. Meanwhile, with the reform of management thoughts and marketing ideas, the content of enterprise informationization construction will be changed, the enterprise not only focuses on the integration of informationization construction inside the enterprise, but also the informationization

construction both inside and outside the enterprise for reaching smooth flow and best effect of internal and external resources.

Single management thought and management mode can no longer meet the requirements for the development of enterprises, and enterprise applications impregnated with modern management thoughts such as ERP, CRM, SCM, BI, knowledge management, and electronic commerce are gradually becoming the enterprise's important tools to help it improve competitive advantages and bring core competitiveness into play for occupying the market. These different enterprise applications contain modern management thoughts belonging to "soul" and also cover information technology belonging to "trunk," both of which are interdependent and indivisible.

It is predictable that the information system on the basis of these management thoughts will gradually enter into the mainstream of information system and integrate with enterprise information system cored by ERP to form an overall informationization solution of supporting the growth of enterprises.

3.2 System Application Networking and Development Platform Standardization

In the age of global economy integration and network economy, rapid development of the Internet and communication technology has thoroughly changed the enterprise's production and management modes and operation methods. The dependence of enterprises on the Internet is as important as that of enterprises on electric power and telephones. There is no such thing as Agile Manufacturing, Virtual Manufacturing, Lean Production, Customer Relationship Management, Supplier Relationship Management, and Electronic Commerce without the application of the Internet. The system based on the Internet is adopted so as to realize group management, management in different areas, mobile office, and global supply chain management.

Computer technology develops up to now, and the closed exclusive system has died out. The following have become the standards that the application system must comply with: the system structure based on B/S; communication protocol; support standard database access; support XML heterogeneous systems interconnection; realize application system independent of hardware platform, operation system, and database; and realize the system's openness, integration, expansibility, and interoperability. Contrarily, the system inconformity with the above standards has no future.

3.3 Business Process Supportable

The business process of the enterprise is dependent on information system more and more. The business process is continuously innovated through the application of information technology, which greatly improves the enterprise's achievements in

indexes such as cost, speed, quality, and service. Traditional ERP lacks effective control and management of business process. When the enterprise's organization structure or business process changes, the enterprise's requirements on data resource sharing is improved, and when the enterprise's business develops, the information system will not be very adaptive, and will even need redesign.

Modern enterprise information system's requirement on supportable scope of business process is further improved with more rich content, including (1) from supporting enterprise internal process to supporting enterprise external process; (2) from supporting isolated process to supporting overall integration of process; (3) from commerce process to cooperation design, research & development, and other processes; and (4) from pure support function to the promotion of business process reengineering (BPR) and business process improvement (BPI). Work flow management technology is an important method to solve the integration of business process, is highly emphasized, and obtains rapid development. The integration of the work flow management technology and ERP or other management information systems will realize the management, control, and automation of the business process; makes true integration of enterprise leaders and business system; and then realizes the reengineering of enterprise business process.

However, the change of business process will inevitably result in the adjustment of organization structure and vice versa, because business process and organization structure are closely interrelated. Through the introduction of information system's support function, originally complex enterprise business process can be simplified, sped up, and collaborated. In addition, many problems existing in the original business process like delay, conflict, error, rework, can be eliminated, which greatly improves working efficiency and working effect of enterprises.

3.4 Application System Integration

Enterprise informationization involves many aspects: technical system informationization, including CAD, CAM, CAPP, PDM, and PLM; management informationization, including ERP, CRM, SRM, BI, and EC; and manufacturing process automatization, including NC, FMS, AS/RS (automated storage and retrieval system), and MES (manufacturing execution system). All these systems serve for enterprise operation strategies and have a great deal of information sharing and exchange among them. On the basis of successful operation of unit technology, they need fulfillment of system integration to maximize the application effect.

3.5 Intelligent Information System

Integration of management information system aims at information exchange and data sharing supported by computer and network technology, database and data warehouse, and data mining technology. Following the global economic integration,

world economy, and the stride of Chinese economy toward knowledge economy, the management information system's architecture and processing capacity should accommodate requirements of knowledge economy. Including all functions of management information system, knowledge information system has its core in expert system. Management information system develops toward intelligent information system which features strong knowledge innovation function, capacity of solving nonstructural matters, leading role in decision making, and guidance for people. It is also an intelligent network information system based on network neuron components and genetic algorithm.

Intelligent information system is designed to solve problems of management information system, such as low intelligence degree, unbalanced human-machine interaction. So, in comparison with traditional management information system, intelligent management information system has many unparalleled advantages in areas where intelligent management is urgently needed. It can be easily fulfilled just by adding expert subsystem with intelligent reasoning on the basis of the traditional management information system. Entering standard conditions, real-time results satisfying such conditions can be obtained, avoiding deviation caused by human factors. That is really scientific and highly efficient. So, intelligent information system has great potentials and advantages in applications.

4 Conclusion

Chinese enterprise information system is experiencing a course of rapid development. Further practice is needed to perfect its management thoughts and methods, architecture, and functions. Information system itself is a complicated human-machine system. Therefore, modern enterprises should choose suitable ways of informationization with their own characteristics when they are constructing and maintaining enterprise information system. Enterprises should design an informationization plan according to the policy as stipulated in 863/CIMS, i.e., overall planning, step-by-step implementation, benefit driven, and key breakthrough. Progress of science and technology is endless, so is enterprise informationization.

References

1. Xue, H. (2003) Management Information System (Fourth Version), Tsinghua University Press, Beijing, pp. 430–434.
2. Chen, Y., and Cai, S. (2005) Research on Development Reasons and Trend of Management Information System. *Business Research*, 14: 4–6.
3. Xiao, K. (2008) Analysis on Development Trend of Enterprise Information System. *China Management Informationization*, 8: 86–89.
4. Yang, Y. (2009) On Development of China Enterprise Information System. *Silicon Valley*, 2: 198.

Asymmetrical Effects of Using Positive and Negative Examples on Object Modeling

Narasimha Bolloju, Christoph Schneider, and Doug Vogel

Abstract Analysis and design is a critical distinction (and consummate challenge) of information systems as a discipline. Object orientation as a mantra and central focus has characterized much recent analysis and design attention. The role of object models in information systems development and challenges associated with developing quality object models especially by novice analysts is a salient point of interest. Illustrative examples of good-quality and poor-quality object models or parts of object models offer guidance available for learning and developing object models. Our research attempts to assess the usefulness of using positive versus negative examples in teaching object modeling skills, so as to enable better learning outcomes for novice analysts. Results of a controlled experiment comparing the effects of positive and negative examples on the quality of object models produced show that positive examples enhanced syntactic quality, negative examples improved semantic quality, and neither had much impact on pragmatic quality.

Keywords Object modeling · Teaching and learning · Instructional design

1 Introduction

Today's tightly connected world has seen rapid advances of information systems used by individuals and business organizations. This progress has fueled rapid changes in information systems development (ISD) methodologies, leading to new and continuing challenges for information systems professionals. In ISD, object models are widely used for capturing information systems requirements in terms of use case diagrams and narratives which are later used to identify classes with attributes and operations and relationships among those classes. With the increasing popularity of UML and object-oriented programming languages, object modeling became an important step in the systems development process. Though object

N. Bolloju (✉)

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

models are considered extremely useful in supporting activities such as client verification, clarifying technical understanding, programmer specifications, and maintenance documentation, even some experienced analysts may not be using these models due to lack of understanding [7]. Consequently, object modeling 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, 16, 17]).

A variety of teaching and study aids, ranging from undergraduate- and graduate-level textbooks (e.g., [3, 6, 8]) to books such as “Object Oriented for Dummies,” have been devoted to helping students understand and internalize the concepts related to object modeling, so as to enable students to apply these concepts when solving hypothetical and real-world problems. By definition, object models have a large visual component, so that they can be understood by a variety of stakeholders. Given this highly visual nature, it is common to include both conceptual knowledge as well as examples when teaching object modeling. A quick perusal of popular textbooks related to object-oriented systems analysis and design reveals that most textbooks use a combination of positive and negative examples for demonstrating the application of object modeling techniques (see, e.g., [8]). However, the effects of using different types of examples are not well understood, and conflicting arguments regarding these effects on learning effectiveness have been presented in various academic fields. Some sources advocate the use of positive examples (e.g., [10, 11]), whereas others argue for the use of negative examples, based on the notion of positive–negative asymmetry [15]; yet others argue for a balance of positive and negative examples (e.g., [1]). Given the dearth of research related to the effects of type of examples on the effectiveness of teaching object modeling, these effects are even less understood in this context.

Our current study attempts to help filling this gap in understanding by focusing on the main research question: *what is the influence of positive versus negative examples on learning object modeling skill?* In an attempt to answer this question, we conducted a laboratory experiment using student subjects to assess the effects of different examples on learning which is measured in terms of the quality of object models created. As the focus of our study is on the use of examples in object modeling, we excluded other mechanisms to enhance learning, such as the provision of positive or negative feedback on the work produced; further, we limited our focus to object modeling or class diagramming. The next section of this chapter provides a review of prior literature and the development of the hypothesis, followed by a description of our methodology. In Section 4, we present the results of our study, and in Section 5 we conclude with a discussion of our findings.

2 Effects of Positive and Negative Instructions on Learning

In education, as well as in other fields, it has long been recognized that concepts and/or skills can be best learned through examples, and scholars have established that a combination of positive and negative examples would be most effective in

concept learning (see [1]). In order to maximize effectiveness and efficiency of instructional materials, researchers have attempted to determine the optimal ratio of positive to negative examples in instructional materials. Reviewing various studies related to this issue, Ali [1] concluded that “instructional materials should *not* include more negative examples” and that the “number of positive examples should be *equal* to or *greater* than the number of negative ones” (p. 6, emphasis in the original).

Contrasting this, however, is the use of negative examples and antipatterns, which have gained much popularity, especially in the field of software engineering, to the extent that entire books have been devoted to these (e.g., [5, 12]). Although there are different definitions of antipatterns, these can loosely be described as pitfalls, examples of “how not to do it,” or what can go wrong in the software development process (see, e.g., <http://www.ibm.com/developerworks/webservices/library/ws-antipatterns/>). As the use of antipatterns, by definition, can be considered a negative teaching mechanism, exaggerated use of antipatterns could have potentially negative effects on learning. Kotzé et al. [10, 11] argued that the use of antipatterns may not be compatible with the internal processes of acquiring and representing knowledge. An empirical study in the context of human–computer interaction, using positive and negative guidelines as substitutes for patterns and antipatterns, respectively, confirmed that students taught using positive examples performed significantly better than students taught using negative examples. As a result, Kotzé et al. strongly cautioned against the use of negative guidelines in the teaching of human–computer interaction concepts and pedagogy in general.

In addition to academic disciplines such as education (and researchers focusing on education within certain disciplines), the effects of negative information have received much attention in the field of psychology. It has been discovered that in various situations, humans react more strongly to negative stimuli than to equally intense positive stimuli. This greater sensitivity to negative information has been referred to as negativity bias, negativity effect, or positive–negative asymmetry, depending on the context and researchers (e.g., [9, 15]). In this view, greater weight or importance is accorded to negative information than to positive information. According to Peeters and Czapinski [15], negative stimuli “elicit more ‘why’ questions” and “lead to more complex cognitive representations” (p. 46). In sum, there are various viewpoints regarding the effects of positive versus negative examples, and it is unclear how far the effects in the literature hold for the teaching of object-oriented concepts.

One way to evaluate effects on learning outcomes is the testing (direct or indirect) of learning aspects. This would allow assigning a grade or a score to an object model – not unlike a grade when evaluating a student’s work. However, a summary grade is unable to capture the different features or elements of an object model as compared to the expected solution(s). To provide a better way of assessing the quality of object models, Lindland et al. [13] proposed a conceptual model quality framework, which has gained wide acceptance through its application in many research studies. This framework applies three linguistic concepts, viz., syntax, semantics, and pragmatics, to assess the quality of conceptual models. Syntactic

correctness of a model implies that all statements in the model are according to the syntax of the language. Semantic quality addresses validity and completeness goals. The validity goal specifies that all statements in the model are correct and relevant to the problem domain. The completeness goal specifies that the model contains all statements about the problem domain that are correct and relevant. Pragmatic quality addresses the comprehension aspect of the model from the stakeholders' perspective. Studying the quality of object models or class diagrams from these three perspectives, Bolloju and Leung [2] identified several commonly occurring quality problems, such as improper naming of classes and associations, missing important classes and associations, incorrect multiplicity specifications for associations and aggregations, poor layout of diagrams, and use of redundant attributes and associations.

By ensuring that commonly occurring quality problems are absent or minimized in those models, it should thus be possible to develop good-quality object models. In this regard, training can be provided to the learners using either a set of positive examples stating the properties of a good object model or a set of negative examples illustrative of specific quality problems or a combination of both. As an illustration, one guideline for enhancing the syntactic quality of class names is that names should be nouns or noun phrases and must start with an uppercase letter. In a teaching situation, a positive example of this guideline can be provided via a sample object model containing a class with the name "Employee" and show that it is correctly named because it is a noun and starts with an uppercase letter. Alternately, a negative example of this guideline can be provided with a class named "employed" that is improperly named as it is not a noun and does not start with an uppercase letter. Appendix 1 provides some sample descriptions of common quality problems along with corresponding positive and negative examples.

If the information content in the positive and negative examples is identical, we should expect that using positive versus negative examples for teaching of object model development should not influence learning outcomes, i.e., the quality of the object models developed should be equal when using the different types of examples. Further, the conflicting arguments in the literature regarding the effect of positive and negative guidelines on learning do not clearly suggest the superiority of one method over the other. Thus, as an initial step toward understanding the specific effects on the three components of the quality of object models, we propose the following hypotheses:

- H1: The type of examples used during training will make no difference in the syntactic quality of object models developed.
- H2: The type of examples used during training will make no difference in the semantic quality of object models developed.
- H3: The type of examples used during training will make no difference in the pragmatic quality of object models developed.

3 Methodology

To test the effects of type of examples on learning outcome, we conducted a controlled study using undergraduate students enrolled in a course on systems analysis. At the beginning of the experiment, each student developed an object model for a journal's (Smallbytes) subscription system (described in Appendix 2); following this, the students were provided with training materials related to common difficulties. One group of students was provided with guidelines associated with positive examples, and the other group was provided with negative examples. These descriptions and examples were incorporated into a training environment (see Appendix 3). After reviewing the training materials, the students were given the opportunity to revise their initial versions of object models. The difference in the quality of the final object models was analyzed to understand possible differences in learning.

The participants in this experiment were undergraduate students of information systems or AMIS major in the college of business of a large metropolitan university in Far East Asia. These students were enrolled in a systems analysis course which mainly covers the object-oriented approach with selected UML techniques. The experiment was conducted toward the end of a 13-week duration course; by that time, the students had become reasonably familiar with object modeling while working on a running case study (related to a university bookstore) in the laboratory sessions and another case study in their project work. The experiment was conducted as an exercise which contributed 5% to their grade.

Four sections of this class participated in the experiment: two sections forming group P (positive examples, $n = 49$) and other two sections forming group N (negative examples, $n = 44$). The profile of these two groups was comparable in terms of their age (19–21 years), the average marks received for the exercise, and the average GPA obtained for this course.

The experiment was conducted during a laboratory session of 100 min duration toward the end of the semester, i.e., after the students had gone through the sessions covering the topic of object modeling and had good practice in the laboratory sessions and their team project work. The sessions included the following activities:

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 Smallbytes subscription system (see Appendix 2) using MS-Visio and submitted the diagram via the Blackboard system (25 min).
3. Break (10 min).
4. Each subject reviewed common quality problems of object models illustrated with either positive or negative examples in the context of a university bookstore

- case study (see Appendix 3 for a sample screenshot of the training environment for group P) (30 min).
5. Each subject then revised their initial version of the object model and submitted version 2 class diagram (25 min).

The assessment of the quality of object models created was based on a three-segment grading scheme. As part of this scheme, the assessment of syntactic, semantic validity, semantic completeness, and pragmatic quality of each object model is estimated after reviewing three different segments of the model using a list of commonly observed quality problems in object modeling which were reported in prior research [2].

Considering the fact that object models created (both versions) could have several variations, it was felt that a three-segment grading scheme could help the graders focus on smaller portions of the object model. Two part-time research assistants, who have already completed a systems analysis course and are in the final year of their undergraduate program majoring in information systems, were trained for assessing the solutions based on the grading scheme. These student helpers graded all 93 pairs of object models and discussed the differences in the quality rating assigned until 100% agreement was reached.

4 Results

Table 1 presents the mean size of the object models created by the subjects in terms of number of classes, total number of attributes, and total number of relationships (generalization, aggregation, and association types). As evident from this table, for both groups, the mean sizes of the revised version of the object models are significantly larger than their corresponding initial versions. A *t*-test of mean sizes across the groups indicated that there is no significant difference in terms of size between the groups P (*n* = 49) and N (*n* = 44) for both initial and revised versions of the object models. Thus, irrespective of the treatment, the revised object models became more complex (and thus possibly more complete).

Table 1 Size of the object models created

Group	Classes per model	Attributes per model	Associations and aggregations per model	Generalizations per model	Total relationships per model
P before	10.43 (2.45)	24.29 (9.98)	5.84 (2.70)	3.04 (2.21)	8.88 (3.80)
P after	12.41 (2.78)	36.22 (9.44)	9.12 (2.27)	3.92 (2.06)	13.04 (3.25)
N before	10.82 (2.53)	22.82 (13.25)	5.41(3.25)	3.41 (2.04)	8.82 (4.05)
N after	12.68 (2.92)	37.84 (9.64)	9.14 (2.43)	4.07 (1.96)	13.20 (3.40)

All the differences between before and after values are significant (*p* < 0.001)

Table 2 Quality of object models by component

Group		Syntactic quality	Semantic quality (comp.)	Semantic quality (validity)	Semantic quality (combined)	Pragmatic quality
P (n = 49)	Before	3.96	2.22	3.32	2.77	4.69
	After	4.32	2.53	3.13	2.83	4.66
	% Change	9.10**	13.76***	-5.64*	2.17	-0.72
N (n = 44)	Before	4.39	2.08	2.93	2.51	4.23
	After	4.49	2.39	3.03	2.71	4.14
	% Change	2.24	14.91**	3.36	7.97**	-2.15

****p* < 0.001; ***p* < 0.01; **p* < 0.05

Table 3 Comparison of group P with group N

Group		Syntactic quality	Semantic quality (comp.)	Semantic quality (validity)	Semantic quality (combined)	Pragmatic quality
Before	P (n = 49)	3.96	2.22	3.32	2.77	4.69
	N (n = 44)	4.39**	2.08	2.93**	2.51**	4.23***
After	P (n = 49)	4.32	2.53	3.13	2.83	4.66
	N (n = 44)	4.49	2.39	3.03	2.71	4.14***

****p* < 0.001; ***p* < 0.01; **p* < 0.05

Each quality component was assessed on a 1–5 scale, based on whether the semantic, syntactic, and pragmatic rules were followed, with 5 being the highest score for each component. Table 2 lists the mean quality for each quality component averaged across the three model segment scores for the two versions of the object models submitted and the percentage change in the quality. This table also shows the percentage change for combined semantic quality.

Table 3 shows a comparison of the different components of the (before and after) models between the two groups. Although the initial versions of group N’s object models were of inferior quality (except for syntactic quality), the quality of revised versions was of comparable quality (except for pragmatic quality).

Hypothesis 1: Usage of positive examples of guidelines has resulted in better syntactic quality of object models compared to using negative examples. As can be seen in Tables 2 and 3, the syntactic quality of the object models, to begin with, is fairly good in both groups. However, especially for group N, it is possible that a ceiling effect (initial versions with quality = 4.39) may have limited the extent of improvement.

Hypothesis 2: The completeness component of semantic quality of object models has improved significantly (see Table 2) for both groups after being exposed to guidelines in the form of positive and negative examples. However, it is interesting to note that the validity component of semantic quality for group P has suffered, i.e., the positive examples contributed *negatively* to this quality component. Hence, the

combined semantic quality for group N has increased significantly (~8%), whereas the difference for group P was not significant.

Hypothesis 3: We did not find any significant difference in pragmatic quality with either positive or negative examples. In fact, there was a small reduction in the mean values of this component. As in the case of syntactic quality, the mean pragmatic quality was relatively high in the initial versions of the object models. The reduction in mean quality could also be a consequence of the increased object model size (or complexity) that made the revised models less comprehensible.

5 Conclusion

This chapter reports the findings from an experimental study aimed at gaining better understanding of the role of positive and negative examples in addressing challenges associated with object modeling. The study included the task of creating object models in an experimental setting and then revising those object models after a review of guidelines using either positive or negative examples. Our analysis of initial and revised versions of the object models using a conceptual model quality framework identified asymmetric effects of the guidelines on different quality components: positive examples enhanced syntactic quality, negative examples improved semantic quality, and neither had much impact on pragmatic quality.

The asymmetry noted with respect to results of positive and negative directions merits special attention. It would be nice to think that students would react best to positive direction. Such is not the case. In fact, being told what not to do, as demonstrated in this research, seems more effective with respect to object modeling. Some of the reasoning is very basic in that positive direction has many possible forms of implementation while negative direction is more focused and specific. In that sense, negative direction is clearer and more memorable. The challenge remains as to how positive direction can be received in a fashion that builds on the educational process and, ultimately, leads to enhanced learning and ability to apply in future contexts.

Some of the limitations of this study include use of students working on a relatively small problem, assessment of object models by trained students who may have failed to recognize equivalent solutions as expected from experts, and ceiling effects on the quality improvement (partially due to the size and complexity of the Smallbytes case study). Further experimentation is required before generalizing these results, especially due to the profile of the subjects (young, oriental) used in the experiment.

Since it is quite common to use examples in teaching and learning of object modeling skill, and the prior research related to the effect of positive and negative examples on learning was inconclusive, the findings from this study are valuable. The asymmetrical effects of using guidelines including positive and negative examples on the quality of object models created can be exploited in instructional design activities that focus on the development of quality object models.

Appendix 1: Some Common Quality Problems and Corresponding Positive and Negative Examples

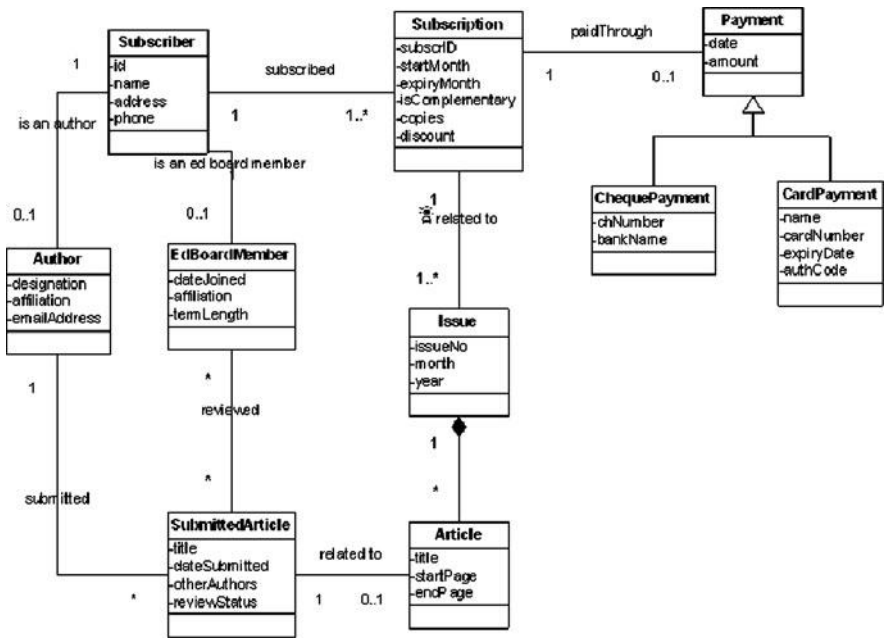
Category	Description	Positive example	Negative example
Syntactic	Class name is a noun, mostly singular, and it begins with an uppercase letter	Class “Sale” is appropriately named using a noun	Class “Sell” is improperly named because it is not a noun
	Association name should be a verb phrase that represents the relationship when it is read along the reading direction from one class to the other	The association name “works at” between “Employee” and “Store” is a verb phrase	The association name “at” between “Store” and “Employee” is not a verb phrase
Semantic	The multiplicity end should have a correct range according to the problem domain description	The multiplicity range for association “places” is correctly specified (a “Customer” places one or more “Orders”)	Multiplicity for association “picked up at” is wrongly specified on the side of class “Order” (i.e., a “Store” can only have zero or 1 “Order” for pickup)
	All attributes relevant to the problem domain entity should be included in the class (i.e., important attributes should not be missing)	The class “Employee” includes many important attributes such as name, phone, and baseSalary	The class “Employee” does not have some important attributes such as name, phone, and baseSalary
Pragmatic	There must be sufficient distinction between (or among) the subclasses (i.e., each subclass has a unique set of attributes and/or relationships)	Sufficient distinction between subclasses “Manager” and “SalesPerson” exists for specialization	Insufficient distinction between subclasses “Manager” and “AssistantManager” for specialization
	Attributes (e.g., age, total amount) which can be computed or replicated from parent side of relationships should not be shown in the conceptual model	No attributes of class “SalesPerson” are redundant	The attribute “storeAddress” in class “SalesPerson” is redundant

Appendix 2: Smallbytes Subscription System Case Study and Expected Solution

Smallbytes 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. The authors receive a year’s free subscription as a token of appreciation for their efforts. Most authors have written only one article during the journal’s 5-year history, but a few have written several. For such authors, another 1-year complimentary subscription is given following the expiry date of the current subscription.

Smallbytes also has an editorial board of advisors, some of whom may also be authors from time to time. The editorial board normally serves for a 1- or 2-year term and they too receive a complimentary subscription to the magazine. The editorial board reviews submitted articles and recommends publication if the article is of good quality.

Smallbytes is sold on a subscription basis. Payments for new subscriptions are normally received by check. Some subscribers pay by credit card. Most subscriptions are for a 1-year period, but the publisher accepts subscriptions for periods longer than or shorter than a year by simply pro-rating the annual subscription price. Most of the subscribers are “single-copy” subscribers; however, some large companies order multiple copies, all of which are sent to the same address. Multiple-copy subscriptions typically involve a small discount from the single-price copy.



Appendix 3: Screenshot of Training Environment Showing a Positive Example (group P)

UML Model Document - Group A Training Material

Properties

Group A Training Material

UML Bookstore for Experiment Group A

Pragmatic

Semantic

Class "Sale" is appropriately named using a noun.

[Syntactic] 03: Class is properly labeled (with a noun)

Multiplicity specification ("1") is correct for "Sale" end of association "made for".

[Semantic] 03: Association multiplicity range is correct

The association name "sold through" between "Book" and "SaleDetail" is correctly named using a verb phrase.

[Syntactic] 04: Association is properly labeled (with a verb)

[Syntactic] 04: Association name should be a verb phrase that represents the relationship when it is read along the reading direction from one class to the other

Multiplicity ("**") is specified for the detail end of aggregation.

[Syntactic] 02: Multiplicity details must be specified for each association

17 Class "Sale" is appropriately named using a noun.
[Syntactic] 03: Class is properly labeled (with a noun)

18 Multiplicity specification ("1") is correct for "Sale" end of association "made for".
[Semantic] 03: Association multiplicity range is correct

19 The association name "sold through" between "Book" and "SaleDetail" is correctly named using a verb phrase.
[Syntactic] 04: Association is properly labeled (with a verb)
[Syntactic] 04: Association name should be a verb phrase that represents the relationship when it is read along the reading direction from one class to the other

20 Multiplicity ("**") is specified for the detail end of aggregation.
[Syntactic] 02: Multiplicity details must be specified for each association

References

1. Ali, A. M. (1981). The use of positive and negative examples during instruction. *Journal of Instructional Development*, 5(1): 2–7.
2. Bolloju, N., and Leung, F. S. K. (2006). Assisting novice analysts in developing quality conceptual models with UML. *Communications of the ACM*, 49(7): 108–112.
3. Booch, G., Maksimchuk, R., Engle, M., Young, B., Conallen, J., and Houston, K. (2007). *Object-oriented analysis and design with applications*, 3rd edition. Upper Saddle River, NJ: Addison-Wesley Professional.
4. Borstler, J., and Hadar, I. (2008). Pedagogies and tools for the teaching and learning of object oriented concepts. *Lecture Notes in Computer Science*, 4906: 182.
5. Brown W.J., McCormick, H.W., Mowbray, T.J., and Malveau, R.C. (1998). *AntiPatterns: refactoring software, architectures, and projects in crisis*. New York, NY: Wiley.
6. Dennis, A., and Wixom, B. H. (2005). *Systems analysis and design with UML version 2.0: an object-oriented approach*. New York, NY: Wiley.
7. Obbing, B., and Parsons, J. (2006). How UML is used. *Communications of the ACM*, 49(5): 109–113.
8. George, J. F., Batra, D., Valacich, J. S., and Hoffer, J. A. (2007). *Object-oriented systems analysis and design*, 2nd ed. Upper Saddle River, NJ: Prentice Hall.
9. Ito, T. A., Larsen, J. T., Smith, N. K., Cacioppo, J. T., and Cacioppo, J. T. (2002). Negative information weighs more heavily on the brain: The negativity bias in evaluative categorizations. *Foundations in Social Neuroscience*, 575–598. The MIT Press.
10. Kotzé, P., Renaud, K., and Biljon, J. (2008). Don't do this – Pitfalls in using anti-patterns in teaching human–computer interaction principles. *Computers & Education*, 50(3): 979–1008.
11. Kotzé, P., Renaud, K., Koukouletsos, K., Khazaei, B., and Dearden, A. (2006). Patterns, anti-patterns and guidelines – effective aids to teaching HCI principles? *Proceedings of the First Joint BCS/IFIP WG13*, 1.
12. Laplante, P. A., and Neill, C. J. (2006). *AntiPatterns: Identification, refactoring and management*, p. 304. Boca Raton, FL: Auerbach Publications.
13. Lindland, O. I., Sindre, G., and Solvberg, A. (1994). Understanding quality in conceptual modeling. *IEEE Software*, 11(2): 42–49.
14. Moritz, S. H., and Blank, G. D. (2005). A design-first curriculum for teaching Java in a CS1 course. *ACM SIGCSE Bulletin*, 37(2): 89–93.
15. Peeters, G., and Czapinski, J. (1990). Positive-negative asymmetry in evaluations: The distinction between affective and informational negativity effects. In: W. Stroebe and M. Hewstone (Eds.), *European Review of Social Psychology* (Vol. 1), pp. 33–60. New York, NY: Wiley.
16. Siau, K., and Cao, Q. (2001). Unified modeling language: A complexity analysis. *Journal of Database Management*, 12(1): 26–34.
17. Siau, K., Erickson, J., and Lee, L. Y. (2005). Theoretical vs. practical complexity: The case of UML. *Journal of Database Management*, 16(3): 40–57.

Part II

IS/IT Project Management

A Social Contract for University–Industry Collaboration: A Case of Project-Based Learning Environment

Tero Vartiainen

Abstract This study determines a social contract for a form of university–industry collaboration to a project-based learning environment in close collaboration with industry. The author’s previous studies on moral conflicts in a project-based learning (PjBL) environment and his 5-year engagement in the PjBL environment are used as background knowledge, and John Rawls’ veil of ignorance is used as a method in the contract formulation. Fair and impartial treatment of actors is strived for with the contract which constitutes of sets of obligations for each party, students, clients, and university (instructors) in the chosen project course. With the contract fair and impartial treatment of actors is strived for and the most dilemmatic moral conflicts are tried to be avoided. The forming of the social contract is evaluated, and implications for research and collaborations in practice are offered.

Keywords University–industry relations · Social contract · Veil of ignorance · Project-based learning

1 Introduction

Successful cooperation between industry and university requires that both parties enter into a thorough-going dialogue about pedagogical and educational thinking [14]. For this collaboration this study offers a social contract approach. Social contract approaches have been studied in business ethics since the early 1980s, and at the heart of social contract thinking is a simple assumption that we understand obligations of key social institutions by attempting to understand what a fair agreement between them would be [6] (p. 38). Taken the university–industry relations there is a need for such a contract as the fundamental responsibilities of the parties differ and because the experience shows that there are conflicts in these relations [17]. The roots for the conflicts may be found in the fundamental responsibilities of industry

T. Vartiainen (✉)
Turku School of Economics, Pori Unit, Turku, Finland
e-mail: tero.vartiainen@oulu.fi

and university and their incommensurableness: The responsibility of university is to do research and teach the truth [2, 10, p. 129], and the responsibility of business is to achieve profitability [4, 5]. Common instantiations of university–industry collaboration are project-based learning environments (PjBL). To define a social contract for such an environment the author entered a project course to identify moral conflicts the parties of the course confront [19–23]. With a moral conflict term the author refers to a morally relevant decision-making situation in which the fulfilment of a moral requirement is at stake. As an example of such a situation, an IT professional was pressured by a client representative to implement a low security level in a system containing sensitive information about the employees of the client organization [1]. The interpretation would be that the IT professional faced two conflicting moral requirements: to implement the client’s demands and to guarantee the confidentiality of the employees’ information. A distinction is made between a moral conflict and a moral dilemma: The former is perceived as resolvable and latter as insolvable (e.g., [12]). The author argues that the more morally conflicting any work system is, the more there is need for discussions, negotiations, and contracting to avoid such conflicts or dilemmas. The results of the author’s previous studies and the author’s practical experience from the same environment show that there indeed is need for a social contract for the particular environment. Pertaining to the fact that PjBL environments have many similarities [7] the contract defined for this particular course most probably develops understanding also in other PjBL environments. In this study the author forms a social contract with John Rawls’ veil of ignorance.

After this introduction the concept of social contract and John Rawls’ veil of ignorance is presented. Then, the author’s findings on moral conflicts perceived by clients, students, and university instructors in a project course are reviewed. After that the process of contract formulation and the contract are presented. Finally, the results of this study are discussed and the limitations are presented.

2 A Social Contract Theory

There are two basic forms of contemporary contract theories [11, p. 188]: Hobbesian contractarianism, which perceives morality as mutual advantage, and Kantian contractarianism, which perceives morality as impartiality. The former recognizes that there is nothing inherently right or wrong concerning the goals we choose to pursue. We do not find any moral values but instead subjective preferences of individuals. The latter form is opposed to this view: It develops the notion of obligation (instead of replacing it like it is in the case of the former). With contracts the inherent moral standing of persons is developed and impartiality is accomplished via making each person’s interests a common concern. Next, a contractarianism-based method to attain impartiality in the contract is presented.

Rawls’ [13] contract theory takes into consideration the conditions when the contract is determined and it aims for genuine equality. The contract is negotiated from a “original position” in which the participants are not aware of facts such as age, religion, level of physical or intellectual ability, economic and social status, religion,

and gender. These facts are “forgotten” as the awareness of them can bias judgment. As a result, under the veil of ignorance, we do not know whether we are poor, rich, white, black, disabled, male, female, young, old, etc. However, behind the veil everyone would know the certain facts on politics, psychology, economics, the existence of social inequalities, and religious beliefs, for example. In other words, under the veil of ignorance we ponder what principles of justice we would like to govern a society we live in, where we could be anyone in any position. Rawls argues in his theory of justice that, in deciding what moral principles they would accept, people under the veil of ignorance would arrange economic and social inequalities in favor of the least advantaged persons. All participants behind the veil also have the right to veto the agreement, which guarantees that the least advantaged parties are taken into account in the contract. Behind the veil, the possibility of becoming one of the least advantaged is open to every participant. Therefore, all would guarantee the positions of the least advantaged by using their right of veto to protect those parties.

Next, the studied project course and the moral conflicts identified by the parties of the course are reviewed to understand the need for a social contract.

3 Moral Conflicts Perceived by Parties of a Project Course

In the project course, groups consisting of five students implement a project defined by a client [19]. The objective of the course is to support students in learning project works skill (e.g., leading, communications) through implementing a project for a real-life client. The clients are typically IT firms such as software houses and IT departments of industrial plants. A client pays the department 8500 EUR for the project and a student group uses 1,375 h planning and implementing the project for the client. The whole project takes 5–6 months to carry out, and each student is expected to practice the job of project manager, ca. 1 month. The contents of a project may vary from design and coding to assessment and research. An example of a project task follows:

The task of the [name of a project group] group is to investigate the usage of EJB (Enterprise Java Beans) in n -layered environment. The goal is to examine the potential uses of EJB in delivering information between the client and the server components. In addition to this, the project group will program a small prototype.

In typical cases, project tasks are ill-defined and need to be redefined during the project. The redefinitions of the tasks and other decisions are made in a steering group. The role of clients is to provide students with guidance on the substance (e.g., technical guidance) and the role of instructors is to guide the process (e.g., planning, reporting). Instructors meet the student groups once a week in a guidance meeting whereas clients and student groups keep contact weekly or even daily basis depending on the task. In this environment the author collected data about moral

conflicts with participant observation, interviewing, diaries, drawing pictures, and questionnaires. Next, the most significant viewpoints of moral conflicts of each party are briefly reviewed.

3.1 Client Representatives Facing the Dirty-Hands Dilemma

Moral conflicts perceived by clients were divided to business-directed and relations-directed conflicts [23]. In business-directed moral conflicts client representatives confront the following three broad questions: (1) How to avoid aiding competitors? (2) What do we get from collaboration with university? (3) How do we benefit from the students' efforts? These business-directed moral problems reflect the fundamental business responsibility, profitability [5], which is the driving force in stockholder theory [15, 16]. Client representatives' loyalty as employees [8, p. 565] makes them adhere to the objectives of their employers and of the owners of their firms (shareholders could be classed as managers' employers). They therefore fulfill their professional duty to uphold profitability, in other words the production of goods and services at a profit [3, p. 303]. Of concern in the business-directed problems was how to get the most out of the students and the collaboration as a whole.

The relations-directed moral conflicts inherent the following three broad questions: (1) How to balance business objectives and social responsibility? (2) How to combine business objectives and objectives of partners? (3) How to attain the business objectives and uphold relations with the individuals of the project? These relations-directed moral problems resemble the idea behind stakeholder theory [15, 16]. Clients still adhere to their objectives, but they also engage in perspective taking, i.e., they try to understand the students' viewpoints, for example, and to find a balance between the beneficial objectives and the students' rights and interests. The following extract exemplifies the issue:

C13: "Yeah, if we think that, well, when there's a client who pays for the project, and he wants to attain certain objectives that have been set for it. If these objectives don't go hand in hand with the objectives that the university has set as educational objectives for the students."

The researcher: "Could you give me an example or tell me somewhat more specifically what this could mean in practice?"

C13: "In practice, it could mean, for example, well, we bypass certain educationally important phases in the project and use professional skills related to the project in the name of efficiency."

This finding bears some resemblance to the dirty-hands dilemma, which is characterized by a conflict between a consequentialist and a principled demand [9, p. 180]: Clients offer the university a real-life learning environment and they opportunistically use the university and the students for their own ends and are ready to violate moral values like honesty (as some client representatives confessed) or otherwise cause harmful consequences for the students (e.g., neglecting the learning objectives, prolonging their studies by employing them).

3.2 Students: The Project Manager’s Morally Complex Job

Students’ moral conflicts are divided to moral failure (deliberation about doing what one perceives to be morally wrong) and moral success (concern for what course of action is morally right in a socially complex situation) related conflicts [21]. In moral failure conflicts there are three broad questions: (1) Will I neglect my obligations in the work tasks? (2) Will I cause harm to others? (3) Will I take advantage at the expense of outsiders? In moral success-related conflicts there are three questions: (1) How to find a balance between conflicting work task-related requirements? (2) How to uphold relations while carrying out the work tasks? (3) How to take into account parties affected by the project? Nearly all these moral conflicts seemed to culminate in the job of project manager. Students in the project manager’s role had to tackle perhaps the hardest moral conflicts related to managing the whole project, the implementation of the project task while at the same time upholding the group members’ motivation. In many cases, the project managers faced situations in which they had to react to moral failure-related conflicts their group members’ confronted: Not all of the group members were equally loyal or committed to the project task or to other members, given the noted avoidance of fulfilling one’s duties and even harassment. Also the abilities of fellow students were for concern for the students in project manager’s job:

“If inside the group there is a person, whom one does not believe to be competent for a task, on what theory one can lean? If one is honest and tells the particular person about it – as a consequence, he either understands the concern or gets hurt. If one does not reveal one’s preoccupations but allocates the task to that person (for example, in a situation in which he is the only one available), it may go wrong, or then again it may succeed. . . . One is not duty bound to blindly trust the other group members. The duty (if we are thinking about the project manager) is to have a good look at the project, to set it in motion with the given resources. If the person described in the previous paragraph is not suitable for the task, one just has to calmly assess the risk one takes in allocating the task to him.”

This finding suggests that the developmental stage of group process [18] in student groups may correlate with the severity and emergence of the moral conflicts confronted by their members. Upholding relations with clients and instructors were also morally conflicting: Honesty and openness issues concerning the real state of the project emerged, for example.

3.3 Instructors: A Job Burdened by Role Strains

The complexity of instructor’s job in the studied project course became visible in the form of role strain, difficulties in meeting the role expectations [22]. Such strains occur on four overlapping levels: organizational objectives, the group’s relations with its instructor, the student’s relations with his or her instructor, and the

instructor's personality. On each level an instructor has to find a proper balance among the diverging expectations. Next, each level is briefly introduced.

Conflicts between organizational objectives between business and university filter down to the instructor's daily work. The instructor may emphasize the two aspects: the learning process (the objective of university) and substance work (objectives of clients) to varying degrees. The "work project" attitude, focusing on fulfilling the client's needs and completing the project task, was common among the students, and it was the responsibility of the instructor to instill the "study project" attitude in them, to encourage them to reflect on the work they had done and to learn from it.

Concerning the group the instructor may be active or passive toward it. The instructor should keep a proper mental distance from the groups in order to make it possible for students to learn independently from their experiences. The more direct guidelines the instructor gives, the more involved he or she is in the functioning of the group to the extent in an extreme case even of becoming a group member. Instructors who have business contacts with the client organization might be more inclined to exhibit this kind of behavior. The leading instructor of the course considered the relations between instructors and student group with the metaphor of riding a dog team:

The leading instructor: "The basic idea of guiding is contained in the model of riding with a dog team . . . the instructor is on a sleigh at the back, shouting hints, which are more or less understood and which steer the team in the right direction."

Taken the individual student viewpoint the instructor should build trust with all the students. It should be possible for the students openly to discuss the real state of their project process (whether in the presence of the whole group or in student-instructor discussions); in that way the instructors are better able to support the group and to make a fair assessment at the end of the course. Providing students with genuine feedback while fostering trust at the same time requires insightful footwork as the following extract shows:

Instructor 9: "We have a responsibility to produce competent people. It is, well, a moral problem related to this educational duty, in a way. Well, we have a sort of societal duty to uphold here. And then, there's a moral problem, well, how you treat these individuals, these students. This is why it's a sensitive thing this project work. These individuals' personality is the object of treatment. . . . In every move you make you have to choose your words and acts carefully."

Taken the personality of the instructor it became clear that for some individual instructors the question of fitting into instructor's job became a hard question. Some instructors confronted hard strains in terms of what is expected from a professional

instructor and how to fit into this kind of job. During these discussions some instructors pondered about finding another job.

4 The Social Contract

4.1 The Process of Defining the Social Contract

The author applied the veil of ignorance keeping in mind that the setup is a socially complicated. The author used the insights from his earlier studies and his 5-year experience as instructor including 2-year period as the leading instructor. Regardless of the acquired knowledge the perspective taking was challenging. The contract was formulated as follows: The author imagined a negotiation in which there were representatives from each party who did not know which party they represent in real life but who would be – after raising the veil – representing a party of the setup. The parties would negotiate about obligations of each party. The parties would consent to and be bound by these obligations once the veil is raised and they start to act in the real world in their roles as clients, students, or instructors. The contract is presented in the next section.

4.2 The Social Contract of a Project Course

The major questions the participants face behind the veil relate to the existence of the collaboration, the existence of each party, and the objectives of each party. To simplify the process the current market economy and the existence of firms and educational institutes as such are accepted as essential parties of society. This being the case the role and objectives of firms and educational institutes have to any way be reflected and discussed behind the veil. Business bases its existence on profitability and therefore each firm has a duty to attain profitability whereas university has the research and learning objectives to be upheld. These fundamental duties became evident in interviews with clients and instructors. Therefore, the contract has to balance these two major duties in a way that all parties accept it. Clients benefit from the collaboration by the results and by employing students whereas students benefit by learning valuable skills. Indeed, it would be agreed that students are the most vulnerable party as they inexperienced individuals and novices in the field. Their position would be safeguarded. Therefore, behind the veil, all parties would agree that clients are expected to concede from a pure business-type client–provider relation to accept risks related to students projects (Table 2, U5) and take into account students' learning objectives. Students' status as novices is also the reason why the tasks should not be business-critical for clients (Table 2, U3). Concerning each particular project collaboration it would be agreed that each one (the university and the firm) should openly describe its objectives concerning the collaboration and accept the fundamental values of the other (Table 2, obligation U1; Table 3, C1).

Table 1 The obligations of students

Toward client	Toward other students	Toward university (instructors)
C1. Fulfill the given project task with reasonable effort	S1. Do your share in the project tasks and reflection tasks	U1. Fulfill the given reflection tasks with reasonable effort
C2. In project managers' job provide client (and the board) truthful assessment of the project status and capabilities of the group to fulfill the project task	S2. Openly discuss about developmental needs of the group	U2. Openly disclose the developmental needs concerning project work skills and skills needed in implementing the project task
	S3. Uphold group spirit	U3. Disclose disturbing behavior to instructors
	S4. In project manager's job, aim to be fair manager and take into account fellow students in managing the work	

Table 2 The obligations of clients

Toward clients (or organization)	Toward students	Toward university (instructors)
C1. Plan in advance how much the firm collaborate with educational institutes	S1. Require reasonably good results from students' work	U1. Openly express the objectives of the organization with respect to the collaboration
C2. Provide representatives of the firm with the resources for collaboration and guidance work	S2. Take into account that students are novices and encourage them to adopt professional practices	U2. Accept the university's role as an educator
C3. Collaborate with other clients of the course if it supports the beneficial objectives of the firm	S3. Provide students with reasonable amount of guidance and constructive feedback	U3. Offer a project task for which there is a real need and which is not business-critical and which is in accordance with the law
	S4. In the case that students are employed to the organization, support their studies	U4. Provide students with real-life project experience by acting as a demanding client
		U5. Accept the risk of project failure

To guarantee students' motivation the project tasks should address real developmental needs in the firm (Table 2, U3; Table 3, C2) – it would not be agreed behind the veil that clients enter the collaboration only with employment objectives (as it happened in the studied case). Both clients and instructors should support students

Table 3 The obligations of university (instructors)

Toward client	Toward students	Toward university (instructors)
C1. Openly articulate the learning objectives and accept the client’s objectives to benefit from the collaboration	S1. Ensure a reasonable infrastructure (space, hardware, software) for the student groups	U1. Assess suitability of prospective instructors for the small group guidance work
C2. Accept clients offering projects that address real needs and which are demanding enough but not business-critical	S2. Support students’ development in learning project work skills	U2. Give reasonable resources for the teaching staff and students
C3. Do not purposely intervene in the markets or invade someone’s territory	S3. Provide possibilities to get project task-related guidance	U3. Openly discuss and share experiences with other instructors
C4. Provide open access to all suitable firms in terms of collaborating with the project course	S4. Reasonably observe the functioning of student groups and intervene when needed S5. Give a truthful assessment	U4. Uphold group spirit among instructors

by providing them with guidance and resources (Table 2, S2 and S3; Table 3, S1, S2, and S3). A significant incentive for client to collaborate with university is to get workforce. If students are employed during their studies, the completion of studies may be prolonged. Therefore, behind the veil, all parties would agree that clients benefiting by employing students should also take into account that students’ needs to complete their studies. Therefore, clients would be obliged to support students’ studies to reasonable extend (Table 2, S4). In what way this would be carried out in practice was left out from discussion. In addition, the objectives of the project and the carrying out of the task should not violate the law (Table 2, U3) by breaking laws on intellectual property rights or on data privacy, for example. Behind the veil each party would agree that the project course is created to provide students with real-life experience. Therefore, the client representatives should act like demanding business clients (Table 2, S1) and require reasonably good results (Table 2, U4), but should take into account the students’ competence level when giving feedback, for example (Table 2, S2). To make the collaboration possible in practice the university should provide the project course with reasonable resources like up-to-date hardware and software (Table 3, U2). However, it is possible that clients provide infrastructure for specific tasks.

Behind the veil the instructor’s relation to its group would be dealt in-depth. Agreement would be attained about the instructor’s role as a coach. There might be an agreement that instructor should not intervene in the process of implementing the task to such extent that the instructor becomes a group member. However, the instructor should support, encourage, and inspire the group and its individuals

(Table 3, S2 and S4). It might be agreed that both students and instructors should consciously build a trustful relation in the way that students could express their learning concerns openly to their instructor (Table 1, U1 and U2). In addition, instructors should provide students with possibilities (e.g., external specialists) to get project task-related guidance in the case that clients are not able to give it (Table 3, S3) and provide students with a truthful assessment (Table 3, S5). Pertaining to the experiences that not all university teachers are suitable for instructors' job recruiting teachers for collaboration should be carefully dealt with (Table 3, U1). Instructors should uphold good spirit in instructors' group as they need professional guidance from each other and the need to trust each other is evident (Table 3, U4). To guarantee reasonably good and truthful assessment (Table 3, S5) instructors should share their experiences and support each other (Table 3, U3).

The inner life of student group would also be considered behind the veil. The experience shows that the major concern for student groups is upholding good spirit and getting the work done. Therefore, it would be agreed that a group member should aim to uphold good spirit in the group (Table 1, S3), do one's share in the project tasks and reflection tasks (Table 1, S1 and S2). Especially, in project manager's job students should aim to be fair managers and take the fellow students into account in managing the work (Table 1, S4). Project manager would also be obliged to provide client truthful information about the state of the project and students' capabilities in carrying out the project task (Table 1, C2). Students would also be obliged to fulfill the given project task with reasonable effort (Table 1, C1). Some students confronted moral conflicts in which they wondered if disturbing behavior of other students should be disclosed to instructors. As the experience shows it happens – very seldom fortunately – that the students engage in harassing behavior toward each other. Behind the veil it would be agreed that such behavior is disclosed to instructors (Table 1, U3).

Finally, coming back to the university–industry relations, the university as the promoter of the collaboration with the local firms should not aim to influence markets or to invade someone's territory by offering "cheap labor" (Table 3, C3 and C4). Participants concealed behind a veil of ignorance would consider the provision of equal opportunities to firms in the surroundings of the university to be fair treatment of business actors. Clients should consciously plan collaboration with educational institutes (Table 2, C1) so that they would be able to provide employees resources for collaboration (C2). Clients taking part in the collaboration the same time might get mutual advantage if they collaborate (C3).

5 Discussion

In this study a social contract was defined for a PjBL environment to attain fair and impartial treatment of actors and for avoiding the most dilemmatic moral conflicts. In the contract formulation veil of ignorance by John Rawls was applied, and as a result obligation sets for students, clients, and university (instructors) toward each others were formulated. The contract recognizes students as the most vulnerable

party and therefore their position is safeguarded. The beneficial objectives of clients are accepted and because of the benefits they are expected to bear the risks relating to student projects. Indeed, the collaboration is perceived to differ from the business client–provider relations because of the low experience level of students and learning objectives of the project course. The university’s role as educator is accepted but they are required to support students in attaining the project task objectives. Students are required to implement the project task as well as it is reasonable to expect to actively learn project work skills and to uphold the group spirit. The relation between students and instructors is critical and therefore both parties are obligated to genuinely build a trustful relation.

5.1 Limitations and Future Research

The current formulation of the contract bases on a single project course in a Finnish university. This contract is the first step toward a social contract of participants in PjBL. Although the generalizability of the contract to other PjBL environments may be questioned the fact that there are similar characteristics in PjBL environments (see for characteristics of PjBL in [7]) the contract in its current form may give insights into other environments. A significant strength and bias in the current research process is the author’s involvement with research and teaching activities the same time at the research stage. On the one hand the immersion is strength as he was able to attain the insider’s viewpoint. On the other hand by becoming an insider one may become blind to issues outsider perceives easier. In addition, there are at least three critical questions pertaining to the results of this study: (1) Is it possible to attain justice and impartiality with the proposed contract? (2) Would the participants of a project course agree to the contract in real life [6]? (3) Would the use of the contract decrease the number of dilemmatic moral conflicts emerging? To answer these questions empirical studies are needed – based on action research, for example.

Acknowledgments I wish to thank the anonymous reviewers for insightful feedback.

References

1. Anderson, R. E., Johnson, D. G., Gotterbarn, D., and Perrolle, J. 1993. Using the New ACM Code of Ethics in Decision-Making, *Communications of ACM* (36)2: 98–107.
2. Brown T.L. 1985. University-Industry Relations: Is There a Conflict? *Journal of Society of Research Administrators* 17(2): 7–17.
3. Buchholz, R.A., and Rosenthal, S.B. 1999. Social responsibility and business ethics. In: Frederick, R.E. (ed.) *A Companion to Business Ethics*. Oxford, UK: Blackwell Publishers. pp. 303–321.
4. Carroll, A.B., 1991. The Pyramid of Corporate Social Responsibility: Toward the Moral Management of Organizational Stakeholders. *Business Horizons* 34(4): 39–48.
5. Carroll, A.B. 1999. Ethics in Management. In: R.E. Frederick (Ed.) *A Companion to Business Ethics*. Oxford, UK: Blackwell. pp. 141–152.

6. Dunfee, T., and Donaldson, T. 1999. Social contract approaches to business ethics: bridging the “is-ought” gap. In: Frederick R.E. (ed.) *A Companion to Business Ethics*. Oxford: Blackwell Publishers. pp. 38–55.
7. Helle, L., Tynjälä, P., and Olkinuora, E. 2006. Project-Based Learning in Post-Secondary Education: Theory, Practice and Rubber Sling Shots, *Higher Education* (51)2: 287–314.
8. Johnson, D.G. 1995. Professional ethics. In: D.G. Johnson, H. Nissenbaum (eds.) *Computers, Ethics, and Social Values*. Englewood Cliffs, NJ: Prentice Hall. pp. 559–572.
9. Kaptein, M., and Wempe, J. 2002. *The Balanced Company, A Theory of Corporate Integrity*. Oxford: Oxford University Press.
10. Kenney M. 1987. The Ethical Dilemmas of University-Industry Collaborations. *Journal of Business Ethics* 6: 127–135.
11. Kymlicka, W. 1991. The social contract tradition. In: Singer, P. (ed.) *A Companion to Ethics*. Oxford: Blackwell. pp.186–196.
12. Nagel, T. 1987. The fragmentation of value. In: C.W. Gowans (ed.) *Moral Dilemmas*. New York, NY: Oxford University Press. pp. 174–187.
13. Rawls, J. 1971. *A Theory of Justice*. London: Oxford University Press.
14. Slotte, V., and Tynjälä, P. 2003. Industry-University Collaboration for Continuing Professional Development. *Journal of Education and Work* 16(4): 445–464.
15. Smith, H.J. 2002. Ethics and Information Systems: Resolving the Quandaries. *The DATA BASE for Advances in Information Systems* 33(3): 8–22.
16. Smith, H.J., and Hasnas, J. 1999. Ethics and Information Systems: The Corporate Domain. *MIS Quarterly* 23(1): 109–127.
17. Stankiewicz, R. 1986. *Academics and Entrepreneurs, Developing University-Industry Relations*. London: Frances Pinter Publishers.
18. Tuckman, B.W. 1965. Developmental Sequence in Small Groups, *Psychological Bulletin*, 63(6): 384–399, American Psychological Association.
19. Vartiainen T. 2005. *Moral Conflicts in a Project Course in Information Systems Education*. Dissertation thesis. Jyväskylä Studies in Computing 49. Jyväskylä: University of Jyväskylä.
20. Vartiainen T. 2006a. Moral problems in industry-academia partnership – the viewpoint of clients on a project course. *Proceedings of the Fifteenth International Conference on Information Systems Development (ISD’2006)*, Budapest, Hungary.
21. Vartiainen T. 2006b. Moral conflicts perceived by students of a project course. In: A. Berglund and M. Wiggberg (Eds.) *Proceedings of 6th Baltic Sea Conference on Computing Education Research, Koli Calling 2006*. Uppsala University, Uppsala, Sweden.
22. Vartiainen T. 2007. Moral Conflicts in Teaching Project Work: A Job Burdened by Role Strains. *Communications of the Association for Information Systems* 20(article 43): 681–711.
23. Vartiainen T. 2009. Moral Problems Perceived by Industry in Collaboration with a Student Group: Balancing between Beneficial Objectives and Upholding Relations. *Journal of Information Systems Education*. Spring Issue, 20(1): 51–66.

Replacement of the Project Manager Reflected Through Activity Theory and Work-System Theory

Tero Vartiainen, Heli Aramo-Immonen, Jari Jussila, Maritta Pirhonen,
and Kirsi Liikamaa

Abstract Replacement of the project manager (RPM) is a known phenomenon in information systems (IS) projects, but scant attention is given to it in the project management or IS literature. Given its critical effects on the project business, the organization, the project team, and the project manager, it should be studied in more depth. We identified factors which make RPM occurrences inherently different and we show that work-system theory and activity theory give comprehensive lenses to advance research on RPM. For the future research on RPM we identified three objectives: experiences on RPM, process model for RPM, and organizational culture's influence on RPM occurrences.

Keywords Project management · Turnover · Project manager

1 Introduction

The project manager is the key to project success [11, p. 575, 20, p. 181, 29, p. 189, 37]. Verner et al. [40] found that the capability of the project manager played an important role in project performance, and according to Nicholas [29] (p. 172), the role is so central that “without it there would not even be project management – the project manager being the glue holding the project together and the mover and shaker spurring it on.” Given the plurality of competences and the managerial complexity [2, 6], see also [8, 18], if the manager needs to be replaced – for whatever reason – poor handling of the replacement may have a variety of detrimental effects on the success of the project, on the team spirit, on the revenues, and on the new manager (for example, cf. [10]). Indeed, Vartiainen and Pirhonen [39] in their exploratory study found that replacement of the project manager (RPM) in an IS context was a critical event: it affected “everything” in the project and brought “chaos” to it. This is particularly the case if it is not professionally handled.

T. Vartiainen (✉)
Turku School of Economics, Pori Unit, Turku, Finland
e-mail: tero.vartiainen@oulu.fi

It is thus clear that RPM is a complex phenomenon and has economic and social consequences (e.g., for the client and the team). In order to add to the research on RPM in the IS context we introduce two theories, social–cultural–historical activity theory [12, 41] and work-system theory [3, 4]. The first of these provides an analytical framework within which to study human activity, and the second offers a systemic approach for a working organization. We also assess and compare the theories with respect to how they interpret RPM and suggest some future research directions they inspire. In applying the theories we take the viewpoint of a project-based organization handling multiple temporary projects.

Following this introduction we briefly discuss the central role of the project manager and the prior literature on RPM. We introduce activity theory and work-system theory in Section 3, and in Section 4 we offer our interpretation of how the theories are applied to RPM and of the factors that make RPM occurrences inherently different. In Section 5 we suggest future directions for research on RPM and evaluate the study.

2 On the Project Manager's Central Role and RPM

The project manager is a key person as he or she manages all the critical functions of the project, including the planning, organizing, staffing, directing, and controlling [15, 24, 26, 35, 38]. Integration encapsulates project management in a single word [43, p. 12], see also [33]: it is the responsibility of the project manager to integrate the variety of equipment, supplies, materials, human resources, and technologies needed to produce the product in conformance with the requirements, on schedule, and within budget. There are four main groups of stakeholders with a vested interest in the activities and results of the project [36, p. 208]: project champions (e.g., investors, clients), project participants (e.g., the project manager, the project team, suppliers), community participants (e.g., political groups, community members), and parasitic participants (e.g., activists, the media). As we see it all these groups comprise the project organization. Project success has different meanings and different degrees of importance to stakeholders, but the goals nevertheless have to be defined and measured [36, p. 219]. Champions and participants have the strongest impacts on success. Moreover, if the interests of the champion are not satisfied the project is perceived to have failed. For these reasons recruiters face a critical decision in selecting the right project manager, someone who is able and willing to function in an autonomous mode and to take full responsibility for the decisions and accept full accountability for the project performance [17, p. 102]. Getting right person for the job is critical given the multitude of challenges to be faced [9, 28, 31, 38] in hypercompetitive and chaotic business environments [21, 44]. These include risk [42] and quality issues, as well as leadership issues [34]. The reasons why projects fail can often be traced back to managerial rather than technical problems [16, 38]. According to Jurison [19], successful projects are based on the project manager's broad experience and managerial and interpersonal skills. In practice it is very difficult to find an experienced and available manager with the right qualifications [38, p. 10], which is why recruiting a successor in the case of

RPM during an ongoing project is a critical task. We find surprisingly few studies on this in the literature.

Abdel-Hamid [1] studied the impact of managerial turnover on software project performance. This study was simulation-based, conducted in a laboratory environment. The results indicate that managerial turnover may lead to a discernible shift in cost/schedule trade-off choices, which affect staff allocation and project performance in terms of cost and duration.

Parker and Skitmore [31] investigated project management turnover and found that its causes were related to career and personal development and dissatisfaction with the organizational culture and the project management role. According to their results the turnover occurs predominantly in the execution phase, and the turnover event negatively affects the performance of the project team and of the project, and possibly the competitive advantage of the organization. Their study was quantitative in nature and the subjects represented project managers from an international aerospace company.

Vartiainen and Pirhonen [39] carried out an exploratory study on RPM, interviewing 10 experienced project managers about what it meant, and what effects it had on projects. They also determined what kind of knowledge should be transferred from the preceding PM to the successor [32]. Most of the subjects referred to replacing the project manager in negative terms: replacement was needed if the project was not going as planned or was facing dilemmas (such as the possibility of the objectives not being met in accordance with the schedule) and trust in the manager had been lost. When trust is lost the client may demand replacement. Similar demands may emerge from inside the project manager's organization or even from inside the team. Trust in the manager may be lost if his or her capabilities and competence do not meet the requirements of the project or if his or her way of working and communicating are perceived as deficient. Problems in the personal chemistry between the project manager and the client representatives also emerged as a reason for replacement. Vartiainen and Pirhonen's findings differ from those of Parker and Skitmore [31], who presented a pre-formed questionnaire to their subjects whereas Vartiainen and Pirhonen used open-ended questions. Dissatisfaction with the organizational culture and the job design did not emerge in the latter study, for example.

The above brief reflection on RPM shows that it is a multifaceted phenomenon and that it should be studied through theories that are capable of identifying the important approaches. We found two promising theoretical approaches to organizations and human activities, which we present next with reference to RPM.

3 The Theories

3.1 Work-System Theory providing a Systemic View of RPM

Work-system theory incorporates both the static and the dynamic view of any system in an organization regardless of whether or not IT is involved and regardless of the size of the organization [3, 4]. The static view identifies basic elements of the work system and the dynamic view focuses on how the system evolves over time

though planned changes and unplanned adaptations. A project could be perceived of as a time-limited work system, the aim of which is to produce something and then to go out of existence [3, p. 46]. We argue that work-system theory offers a powerful lens through which to understand RPM from the viewpoint of a project organization. The project manager is the key person in managing the whole project, and therefore when RPM occurs the successor has to take control of all the elements included in the theory. From the static perspective the theory consists of the following elements.

Business processes describe the work performed within the work system. This work could be summarized in terms of one or more business processes in which the steps may be defined tightly or may be relatively unstructured. The activities in each one include information processing, communication, and decision making. *The participants* are the people who perform at least some of the work in the business process. Some of them may use IT extensively, whereas others may use little or no technology. *Information* comprises the codified and non-codified information used and created as the participants perform their work. It may or may not be captured on a computer. *Technologies* include the tools (projectors and spreadsheet software, for example) and techniques (such as management by objectives and optimization) that the system participants use. *Products/services* represent the combination of physical things, information, and services that the work system produces, including any physical or information products, services, intangibles such as enjoyment, and social products such as arrangements and agreements. *Customers* are people who receive direct benefit from the products/services the work system provides. They may be external, receiving the organization's products/services, or internal in the case of employees and contractors. *The environment* includes the organizational, competitive, technical, cultural, and regulatory contexts within which the work system operates. It affects the system performance although the system is not directly dependent on it. *The infrastructure, in turn*, includes the technical and human informational resources on which the work system relies, as well as support and training staff, and shared databases, networks, and programming technology. *Strategies* explain why the work system operates as it does and may relate to the organization or the system.

From the dynamic perspective, the work-system life cycle model incorporates the following phases [3, p. 47]: *Initiation*, which involves clarifying the reasons for changing the work system and what the changes will entail and identifying the people and processes that will be affected; *Development*, which is the process of defining, creating, or obtaining the tools and resources that are needed before the change can be implemented in the organization; *Implementation*, during which the desired changes are made operational; and *Operation and maintenance*, during which the work system is made to operate efficiently. This final phase continues until major changes are needed, at which point a new iteration of these four phases starts. Each of the above phases allows for planned and unanticipated changes. With regard to the implementation and operation/maintenance phase Alter [4, p. 95] recognizes "Unanticipated adaptations," whereas in the initiation and development phases there are "Unanticipated opportunities." RPM used to rescue a troubled project is

indeed unanticipated, and as the findings of Vartiainen and Pirhonen [39] suggest, is connected to many of the static elements described above.

3.2 Activity Theory and the Need for Expansive Learning in RPM

Activity theory distinguishes between temporary, goal-directed actions and durable, object-oriented activity systems (Fig. 1) [12, 23, 41]. The case of project management concerns the latter. In this context the “activity” has a broader meaning than “action” or “operation” (consider a football game as an activity and kicking a ball as an action, for example). Here the activity is the project as a whole. As applied in activity theory the concept of activity means linking events to the contexts within which they occur [7]. The process of the creation, use, and utilization of knowledge in an organization is not a spontaneous phenomenon. According to socio-cultural, historical activity theory there has to be a triggering action, such as the conflictual questioning of the existing standard practice in the system, in order to generate expansive learning [12, 30]. In this study RPM could be considered the triggering action. Expansive learning produces culturally new patterns of activity, and the object of the learning activity is the entire system (here the project) in which the learners (here the project members and manager) are working [13]. Figure 1 illustrates the systemic structure of collective activity according to Engeström [12].

This study adopts the idea that the problem with management decisions often lies in the assumption that orders to learn and to create new knowledge are given from above [12]. The enabling of knowledge sharing is required in order to generate new knowledge in the organization. In the case of RPM there is either an external or an internal need for learning in the entire activity system (e.g., the project), which includes the new project manager. The external triggering action may be value conflict with stakeholders, for example, and the internal triggering action could be the project manager’s lack of experience and competence, or conflict within the project organization (personal chemistry).

Engeström [12] suggests that the motivation to learn is embedded in the connection between the outcome and the object of the activity. The object of the collective activity (e.g., the project plan) is transferred to the practical outcome

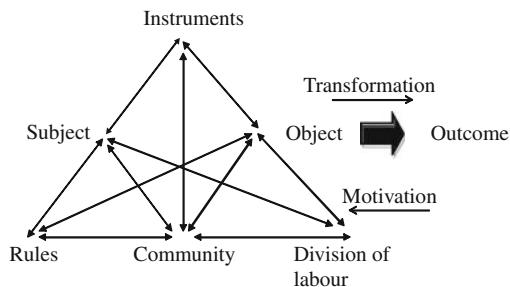


Fig. 1 Systems of collective activity, adapted from [12, p. 962]

(e.g., an information system) (Fig. 1). Achieving practical results through this transformation creates the motivation to change. Findings from research conducted among experienced project managers have confirmed that there is the motivation to share knowledge, but paradoxically there is very little evidence of practical knowledge sharing in the project organization [22]. Therefore it could be argued that there is a need for modeling action patterns such as RPM in order to ensure knowledge diffusion in the activity system of the project.

In the case of RPM the project organization has to effect transformations that are not yet in place. In other words, it has to learn and operate simultaneously. In practice RPM places the project group in a new social network situation. Traditional learning theories define organizational learning as a process of detecting and correcting errors (e.g., single- and double-loop learning [5]). This tradition has little to offer in an RPM situation. On the other hand, the theory of expansive learning at work (based on activity theory) produces new forms of work activity [13]. An essential component of such learning is shared knowledge, which accumulates in the explicit form of rules and instruments (artefacts and tools), for example, and in the tacit form of cultural-, historical-, social-, experience-based knowledge (Fig. 1). This knowledge, which is tacit in nature, makes the new project manager very dependent on the project's activity system. Next, we apply these theories to RPM.

4 Application of the Theories to RPM

The project manager is a key participant in the project process (Fig. 2). We therefore describe him or her as his or her own entity, set apart from the “participants” (in work-system theory). When the project manager is replaced with a successor manager all the links between the role and the critical issues (all the elements in Fig. 2) have to be maintained or even re-created. We applied both work-system theory and activity theory to the IS project context. Table 1 summarizes what each element means in this context.

According to our interpretation the two theories provide insights that will enable future research to enhance understanding of, explain, and even predict RPM. Its causes most probably lie in the relation between the elements of the two theories and the project manager. In the following we further reflect on RPM as inspired by these theories on the basis of two extracts from the interviews with experienced project managers. Our reflection is based on our practical experience and our project studies in close collaboration with industry. The extracts are adapted from [39].

The authors asked the project managers in an open-ended question to describe what came to mind about RPM:

PM3: “Mainly a situation where the project manager does not enjoy the trust of the client, the steering group or the client’s staff. They do not trust his or her competence or way of acting as a project manager, or perhaps he or she is too inexperienced, or then a more experienced manager might be wanted. The situation has changed or has somehow gotten out of hand.”

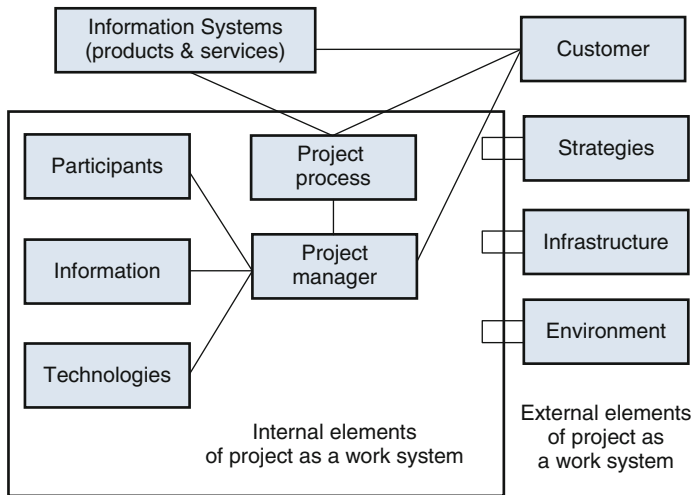


Fig. 2 Elements of work-system theory (adapted from [3, 4]) to the project context

PM2: “Well, it can be a consequence of the fact that the outcomes do not correspond to the requirements or the project does not run-on time and these things cause these chemistry concerns. For example, the client and the project manager might not get along with each other and then it is time to replace the project manager.”

The above extracts incorporate the main elements of the two theories: the project manager, the client, the client’s staff, and the outcomes. Although RPM is described very briefly, its complexity in the project organization becomes clear when alternative situations are taken into account. Bearing in mind the elements of both theories, we identified the following factors that affect RPM and make its occurrence in project-based organizations inherently different:

- A project-based organization as a whole could be perceived of as a work system incorporating many temporary projects (work systems). Organizational cultures [27] in project-based organizations most likely differ, as do the number of projects led by a project manager.
- Project types differ and RPM may occur in different phases.
- The role of the project manager may differ in that he or she may have more of an administrative role or may be more or less involved in the implementation.
- The cultural and historical background and experience and the competence levels of the outgoing and successor managers also affect RPM.

Given this inherent complexity in the RPM phenomenon we need a comprehensive theory in order to understand it fully. We believe that work-system theory and activity theory serve this purpose. The core idea in activity theory is to study the

Table 1 Elements of the theories applied to the context of IS projects

Theory	Element level	Element	Application of the element to IS project contexts
	Internal elements	Participants	People who are active in the project (e.g., project manager, team members)
		Information	Codified and non-codified information used in project work (e.g., user stories, project manager's, and the customer representative's tacit knowledge, project member's personal notes)
		Technologies	Tools (e.g., project management software, programming language and techniques (e.g., walk-throughs, use of user stories)
		Business process	Project processes (e.g., contracting, managerial processes)
		Customer	People who benefit from the project results (e.g., users in the customer organization, the owner of business process)
		Strategies	Project organization strategies; project strategies (e.g., project portfolio)
		Infrastructure	Technical, human, and informational resources that the project relies on (e.g., hardware, support staff, intranets)
		Environment	Organizational, competitive, technical, cultural, and regulatory environment of project (e.g., variety of operating systems, spoken languages, and cultural differences, laws on data registration)
		Products/ services	The combination of artifacts and services the project produces (e.g., information system, maintenance service, user training)
		Subject Rules	Project manager Rules of project management (e.g., norms in communication practices and walk-throughs)
Work-system theory	External elements	Community Division of labour	The project team, the project organization Role definition, task allocation, customer, and end-user involvement (e.g., project manager and secretary selection, participatory design, on-site customer)
		Instruments	Tools, signs, symbols (e.g., project management tools, software, GANTT chart, iterative techniques)
		Object	Contracts, project plans, definition of software product (e.g., requirements definition)
Activity theory	Activity system	Transformation	Outcome An information system, trained users

community (here the project organization) and the subject (here the project manager) in a new pattern of activity (here RPM). This new pattern of activity can only exist through expansive learning in the whole work system (here the project organization). In addition, RPM inevitably engages the project organization (at the supplier's and the customer's sites) in a change management situation. Therefore we argue that these two theories give us a powerful theoretical framework within which to study RPM in more depth.

5 Discussion

In this study we introduced activity theory and work-system theory for the purpose of studying RPM and specified how the elements of these theories relate to the project management environment. We also identified some factors that make RPM occurrences inherently different. We showed that the two theories are applicable for future studies on RPM. The strength of activity theory is that it describes learning and change in organizations, and in our application it describes the dynamics between the project manager and the project organization. Work-system theory was developed in the IS context, and it describes the project structure as a work system. It includes IS elements that are not considered in activity theory. In sum, we suggest that these theories provide promising analytical frameworks for future research on RPM.

First, the RPM phenomenon should be fully understood. In order to attain this goal RPM types and their characteristics (characterized according to the reasons for it and its consequences for project champions and participants, for example) should be determined, and individuals' (participants such as project managers and senior managers) experiences should be studied in terms of how RPM affects the project manager's professional identity, for example.

Second, there is a need to design a process model for RPM. We argue that such a model should take into account both the leadership (e.g., human and social issues) and the management (e.g., concern for production) perspectives (cf. [28, p. 264]). Recruiting a suitable successor manager to serve the needs of the whole project (cf. the competence perspective in [25]), transferring knowledge from the former to the successor manager (cf. Table 1), and ensuring support for both of them (cf. performance management in [14]) are all part of this process.

Third, there is need to reflect on RPM from the perspective of the organizational culture and the system of shared values and beliefs it represents. We argue that it should support project managers facing RPM situations.

5.1 Evaluation of the Study

This study is limited to the IS field, but the general nature of project management may make our interpretations applicable to other areas. As far as the validity of our interpretations is concerned, it is worth noting that three of the authors of this

chapter have industrial experience of project management, and the other two have been involved in project studies in collaboration with industry. They are all engaged in research on project management.

References

1. Abdel-Hamid, T. K. (1992) Investigating the Impacts of Managerial Turnover/Succession on Software Project Performance, *Journal of Management Information Systems*, 9(2): 127–144.
2. Aladwani, A. M. (2002) IT Project Uncertainty, Planning and Success – An Empirical Investigation from Kuwait, *Information Technology & People*, 15(3): 210–226.
3. Alter, S. (1999) A General, Yet Useful Theory of Information Systems, *Communications of the Association for Information Systems*, 1(13).
4. Alter, S. (2002) The Work System Method for Understanding Information Systems and Information Systems Research, *Communications of the Association for Information Systems*, 9: 90–104.
5. Argyris, C. and Schön, D. (1978) *Organizational Learning, A Theory of Action*, Reading, MA: Addison-Wesley Reading.
6. Blackburn, S. (2002) The Project Manager and the Project-network, *International Journal of Project Management*, 20(3): 199–204.
7. Blackler, F., Crump, N. and McDonald, S. (1999) Managing Experts and Competing through Innovation: An Activity Theoretical Analysis, *Organization*, 6(1): 5–32.
8. Birdir, K. (2002) General Manager Turnover and Root Causes, *International Journal of Contemporary Hospitality Management*, 14(1): 43–47.
9. Boddy, D. (2002) *Managing Projects: Building and Leading the Team*, Harlow, Essex: Prentice Hall.
10. Brown, M. C. (1982) Administrative Succession and Organizational Performance: The Succession Effect, *Administrative Science Quarterly*, 27(1): 1–16.
11. Cleland, D. (1984) *Matrix Management Systems Handbook*, New York: Van Nostrand Reinhold.
12. Engeström, Y. (2000) Activity Theory as a Framework for Analyzing and Redesigning Work, *Ergonomics*, 43(7): 960–974.
13. Engeström, Y. (2001) Expansive Learning at Work: Toward an Activity Theoretical Reconceptualization, *Journal of Education & Work*, 14(1): 133–156.
14. Foot, M. and Hook, C. (2008) *Introducing Human Resource Management*. Harlow, Essex: Prentice Hall.
15. Görög, M. and Smith, N. (1999) *Project Management for Managers*, Pennsylvania: PMI.
16. Hartman, F. and Ashrafi, R.A. (2002) Project Management in the Information Systems and Information Technologies Industries, *Project Management Journal*, 33(3): 5–15.
17. Hobbs, B. and Menard, P. (1993) Organizational Choices for Project Management. In: Dinsmore, P. C. (ed.), *The AMA Handbook of Project Management*, pp. 81–108. New York: Amacom.
18. International Project Management Association (2009) *IPMA, Competence baseline*. Retrieved April 12th 2009 from <http://www.ipma.ch/Pages/default.aspx>.
19. Jurison, J. (1999) Software Project Management: The Manager's View, *Communications of the Association for Information Systems*: 2: Article 17.
20. Kezsbom, D. S., Schilling, D. L., and Edward, K. A. (1989) *Dynamic Project Management: A Practical Guide for Managers and Engineers*, New York: Wiley.
21. Kloppenborg, J. and Petrick, T. (1999) Leadership in Project Life Cycle and Team Character Development, *Project Management Journal*, 30(2): 8–14.
22. Koskinen, K. U. and Aramo-Immonen, H. (2008) Remembering with the Help of Personal Notes in a Project Work Context, *Managing Projects in Business*, 1(2): 193–205.

23. Kuutti, K. (1995) "Activity Theory as a Potential Framework for Human-Computer Interaction Research" in Context and Consciousness. In: Nardi B. (ed.) *Activity Theory and Human Computer Interaction*, pp. 17–44. Cambridge: MIT Press.
24. Levine, H. A. (2005) *Project Portfolio Management – A Practical Guide to Selecting Projects, Managing Portfolios, and Maximizing Benefits*, San Francisco: Jossey-Bass a Wiley Imprint.
25. Liikamaa, K. (2006) Piilevä tieto ja projektipäällikön kompetenssit (In English: Tacit Knowledge and Project Managers' Competencies), Dissertation, Publication 628, Tampere University of Technology, Tampere: Tampereen yliopistopaino Oy.
26. Lock, D. (1994) *Gower Handbook of Project Management*, Aldershot: Gower.
27. Lucas, L. M. and Ogilvie, D. T. (2006) Things Are Not Always What They Seem. How Reputations, Culture, and Incentives Influence Knowledge Transfer, *The Learning Organization*, 13(1): 7–24.
28. Maylor, H. (2003) *Project Management*. Harlow, Essex: Prentice Hall.
29. Nicholas, J. (1994) *Managing Business and Engineering Projects: Concepts and Implementation*, Englewood Cliffs, NJ: Prentice-Hall.
30. Nonaka I., Reinmoeller, P., and Senoo, D. (1998) Management Focus. The 'ART' of knowledge: Systems to Capitalize on Market Knowledge, *European Management Journal*, 16(6): 673–684.
31. Parker, S. K. and Skitmore, M. (2005) Project Management Turnover: Causes and Effects on Project Performance. *International Journal of Project Management*, 23(3): 205–214.
32. Pirhonen, M. and Vartiainen, T. (2007) *Replacing the Project Manager in Information System Projects: What Knowledge Should be Transferred?* In: *Proceedings of the 13th the Americas Conference on Information Systems (AMCIS), Reaching New Heights*, August 9–12, Keystone, Colorado, [CD-ROM].
33. PMI 2000, *PMBOK, A Guide to the Project Management Body of Knowledge*, Project Management Institute, Pennsylvania, USA.
34. Smith, G. R. (1999) Project Leadership: Why Project Management Alone Doesn't Work. *Hospital Materiel Management Quarterly* 21(1): 88–92.
35. Thayer, R. (1987) Software Engineering Project Management: A Top-Down View. In" Thayer, R. (ed.) *Tutorial: Software Engineering Project Management*, pp. 15–53. Los Alamitos, CA: IEEE Computer Science Press.
36. Tuman, J. (1993) Models for Achieving Project Success Through Team Building and Stakeholder Management. In" Dinsmore, P. C. (ed.) *The AMA Handbook of Project Management*, pp. 207–223. New York: Amacom.
37. Turner, R. (1999) *Handbook of Project-Based Management, Improving Processes for Achieving Strategic Objectives*, London: McGraw-Hill Companies.
38. Turner, R. (2003) *People in Project Management*, Aldershot: Gower.
39. Vartiainen, T. and Pirhonen, M. (2007) *How is Project Success Affected by Replacing the Project Manager?* In: Magyar, G., Knapp, G., Wojtkowski, W. Wojtkowski, W. G., and Zupancic, J. (eds.), *Advances in Information Systems Development, New Methods and Practice for the Networked Society*, Vol. 2, pp. 397–407. New York: Springer.
40. Verner, J. M., Overmyer, S. P., and McCain, K. W. (1999) In the 25 Years since The Mythical Man-Month. What Have We Learned About Project Management? *Information and Software Technology*, 41: 1021–1026.
41. Vygotsky, L. (1986) *Thought and Language*, Boston, MA: Massachusetts Institute of Technology.
42. Wallace, L. and Keil, M. (2004) Software Project Risks and Their Effect on Out-comes, *Communications of the ACM*, 47(4): 68–73.
43. Webster, F. M. (1993) What Project Management Is All About. In: Dinsmore, P.C. (ed.) *The AMA Handbook of Project Management*, pp. 5–17. New York: Amacom.
44. Yeo, K. T. (2002) Critical Failure Factors in Information Systems Projects. *International Journal of Project Management*, 20(3): 241–246.

Integrating Environmental and Information Systems Management: An Enterprise Architecture Approach

Ovidiu Noran

Abstract Environmental responsibility is fast becoming an important aspect of strategic management as the reality of climate change settles in and relevant regulations are expected to tighten significantly in the near future. Many businesses react to this challenge by implementing environmental reporting and management systems. However, the environmental initiative is often not properly integrated in the overall business strategy and its information system (IS) and as a result the management does not have timely access to (appropriately aggregated) environmental information. This chapter argues for the benefit of integrating the environmental management (EM) project into the ongoing enterprise architecture (EA) initiative present in all successful companies. This is done by demonstrating how a reference architecture framework and a meta-methodology using EA artefacts can be used to co-design the EM system, the organisation and its IS in order to achieve a much needed synergy.

Keywords Environmental management · Enterprise architecture · IS management

1 Introduction

History has shown that the continued existence of businesses depends not only on their economic sustainability but also on their impact on the natural environment and the way they treat their workers. This basic truth was emphasised by Elkington's [1] triple bottom line (TBL) approach to business sustainability: one must achieve not only economic bottom-line performance but also environmental and social accomplishment. Blackburn [2] compares economic sustainability to air and environmental and social sustainability to food: the first is more urgent but not more important than the second. The "2Rs" (respect for humans and judicious resource management) are another essential component of overall sustainability of the business – hence,

O. Noran (✉)
School of ICT, Griffith University, Nathan, QLD, Australia
e-mail: o.noran@griffith.edu.au

a successful enterprise must take a whole-system approach to *sustainable development* [3]. This chapter focuses on the challenges presented by the integration of the environmental sustainability aspect in the business and proposes a solution addressing these challenges based on an EA approach.

2 Tackling Environmental Management Integration

To date most EM efforts are rather disjointed, i.e. specific to business units and not properly supported by the IS and the ICT infrastructure. This means that (a) the company loses coherence as different units approach environmental sustainability in different levels of detail and at a different pace, (b) there is a possible loss of combined or aggregate capabilities due to the various departments not “understanding” each other’s approach to sustainability and (c) top management cannot effectively use the information generated by the environmental reporting functions due to language, format, level of aggregation, etc.

Strategic integration of EM is only achievable if the necessary information is quickly available and is of high quality [4]. The information must be at the fingertips of managers in the form and level of aggregation they need as agility is not compatible with delays incurred by digging out and filtering suitable information for each request. The EM project must be integrated with changes in the enterprise’s IS for effective access to environmental information (constraints, impact, etc.) facilitating the decision-making process [4, 5]. Meeting such challenges requires setting up an EM project with

- (a) top management support for the project champion(s);
- (b) sufficient authority and appropriate human/infrastructure resources;
- (c) a suitable strategy integrated in the general company strategic direction;
- (d) a cross-departmental approach, horizontally and vertically.

These prerequisites are instrumental for the project to trigger organisational culture change, so as to determine *permanent* changes in the way people do things. The above-mentioned requirements match to a good extent the scope of typical enterprise architecture (EA) projects; it is therefore proposed by this chapter that EA may provide a solution to an integrated, coherent approach to the introduction of environmental aspects in the management and operation of all business units. This is desirable because a company whose architecture includes EM, competencies and responsibilities in an integrated fashion will have the necessary *agility* and *preparedness* to cope with the challenges brought about by climate change, thus turning a potential weakness into a strength. Hence, changes in the economic and natural environment will produce less knee jerk, interventionist management behaviour and organisational turbulence.

The EM project would involve several typical steps, such as: identifying the business processes and understanding their impact on the environment (the AS-IS), defining a vision and concept(s) for the future state (the TO-BE), eliciting and

specifying requirements to reach the selected TO-BE state, (re)designing the processes, policies and often the entire organisation according to these requirements, implementing them, continually monitoring the effects and applying some of the previous steps for correction and enhancement. These phases reflect the continuous improvement Plan-Do-Check-Act cycle [6].

3 Environmental Management Artefacts: A Brief Analysis

Companies typically address the mandated and/or perceived requirement to introduce environmental responsibility in their business units by attempting to implement some type of environmental reporting and environmental management system (EMS).

While an EMS is a step in the right direction, when implemented in isolation it will not trigger the cultural change necessary to make environmental responsibility “stick”. Some authors [7] argue that the implementation of an EMS alone (especially if imposed on the organisation) is irrelevant in the absence of a *real* commitment to environmental improvements. Relevant regulation does not help; for example, ISO 14001:2004 [8] only requires that an EMS be designed in such a way that companies can work towards the goal of regulatory compliance and seek to make improvements, not that the company actually achieves environmental excellence or even full compliance with existing laws!

Various reference models (frameworks, methods, etc.) and alternatives to EMS design have emerged. For example, Blackburn [2] proposes a “Sustainability Operating System” – in fact, a management method to achieve sustainability based on the Brundtland report [3], the “2Rs” and the TBL approach applied to sustainability. Willard [9] also recommends a TBL-based approach encompassing economy/profit, environment/planet and equity/people with seven benefits: easier hiring and retention, increased productivity, reduced manufacturing/commercial sites expenses, increased revenue/market share and reduced risk. Clayton and Redcliffe [10] propose a systems approach to integration of sustainability aspects into the business and define the concept of environmental quality as capital (and thus the feasibility of “tradable pollution”).

EM frameworks aim to provide a structured set of artefacts (methods, aspects, reference models, etc.) specialised for the EM area. Some examples are The Natural Step (TNS) Framework, using a systems-based approach to organisational planning for sustainability [11], The Natural Edge Project [12] which proposes a holistic approach (Whole System) taking into account system life cycle and the Life Cycle Management Framework for continuous environmental improvement [13].

Assessment and reporting frameworks aim to assist the measurement and reporting functions of the EMS. For example, the Life Cycle Assessment (LCA) method measures the environmental impacts of products or services relative to each other during their life cycles [14]. The Global Reporting Initiative’s sustainability reporting framework [15] contains reporting principles, guidance and standard disclosures potentially applicable to all types of businesses.

International Standards also cover the EM issue. ISO 14000:2004 is a set of reference models for setting up EMS-s and for life cycle assessment, environmental auditing of processes, environmental labelling and environmental performance evaluation. ISO 14001:2004 deals specifically with EMS-s, aiming to provide a framework for a holistic and strategic approach to the organisation's environmental policy, plans and actions [8]. Standards provide a good starting and reference point for design and assessment; however, as mentioned current EM standards do not define EM performance levels that the company should meet.

Many of the above-mentioned artefacts recognise the need to analyse the life cycle of the products. However, in reality it is typically required to also take into account other life cycles – e.g. those of the host company, of its IS, of the projects set up to (re)design the IS and create the EMS and especially of the EMS itself – and analyse the interactions between these entities in that context. This approach provides a holistic perspective, allowing to represent and understand the business, the relevant projects, the target EMS, its impact on the IS and to identify potential problems and aspects that may not be otherwise obvious. Frameworks describing systems during their entire life (not just at particular points in time), also called *life cycle architectures*, are commonly used in EA.

4 Enterprise Architecture Frameworks, GERAM and GERA

Enterprises are highly complex systems. Therefore, sets of models (sometimes aggregated in *architectural descriptions* corresponding to *viewpoints* representing stakeholders [16]) are produced using various languages in order to control this complexity and allow the enterprise architect and other stakeholders to focus on various aspects of the business. Other types of artefacts commonly used to structure knowledge in EA practice are modelling frameworks (MFs), methods, reference models, ontologies, meta-models, glossaries, etc.; they are typically organised in architecture frameworks (AFs), some of which have underlying meta-models formally describing their structure. Currently there are several mainstream AFs, generic (e.g. PERA [17], TOGAF [18]) or aimed at various domains such as manufacturing (CIMOSA [19], ARIS [20]), defence (DoDAF [21]), information systems [22], etc.

In this research we have selected a reference framework obtained by generalising other AFs and thus potentially expressive enough to contain all the elements necessary to achieve environmental management integration using EA artefacts. This AF is GERAM (Generalised Enterprise Reference Architecture and Methodology), described in ISO 15704:2000. GERAM has been used in practice to guide EA projects [23–25], to assess other enterprise AFs [26–29] and to build a structured repository of AF elements for a project management decision support system [30]. For more details on GERAM see [31].

As can be seen from Fig. 1, the main component of the Reference Architecture of GERAM (called GERA) is a MF containing an extensive set of aspects including life cycle, management, organisational, human and decisional, corresponding to various

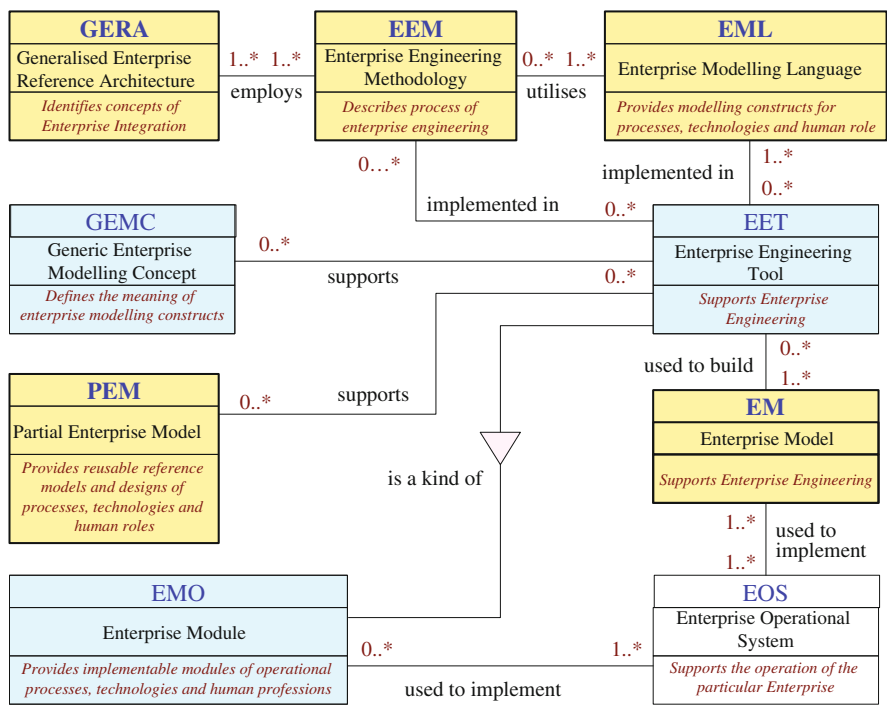


Fig. 1 A high-level meta-model of GERAM (based on [31])

stakeholder concerns [16]. A subset of GERA has been used as a modelling formalism in the creation of a life cycle-based business model as subsequently shown in this chapter.

5 A Meta-methodology for Enterprise Architecture Projects

This chapter argues that EA can provide an overarching and life cycle-based approach in setting up and operating an EM project aiming to produce an EMS in an integrated and coherent manner in relation to the host organisation and other relevant external entities. To illustrate this approach, the researcher has used a *meta-methodology*, or a “method to build methods” applicable for specific types of EA tasks (projects), based on an original approach abiding by EA principles. The meta-methodology, first defined in [32, 33] and tested in several case studies [34–36], employs a set of steps and of sub-steps applicable to each step, as shown in Fig. 2.

In the first step, the user is prompted to create a list containing entities of interest to the project in question, including project participants, target entities (organisations, other projects) and importantly, the project itself. The second step comprises the creation of business models showing the relations between the previously listed

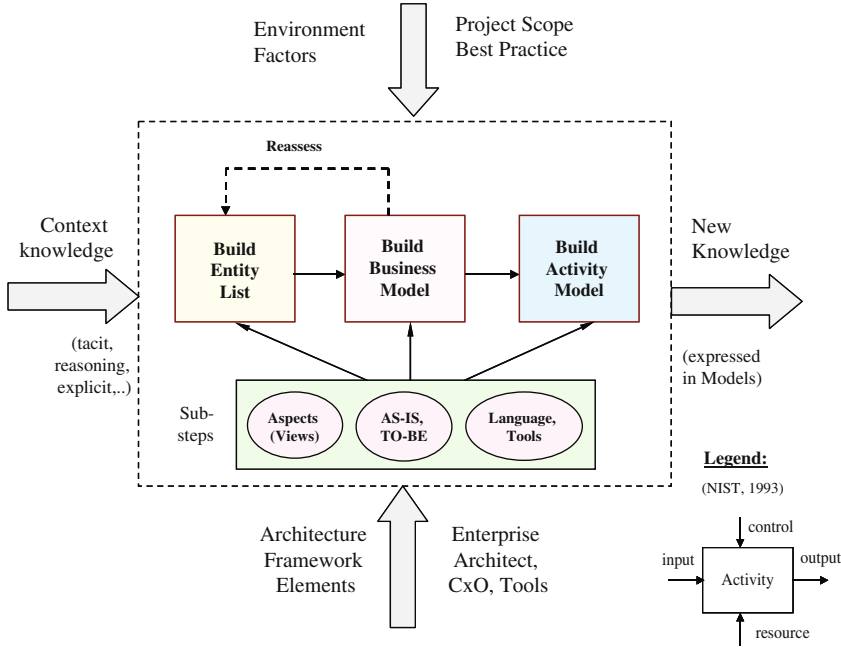


Fig. 2 Meta-methodology concept [30]

entities *in the context of their lifecycles*, i.e. illustrating how entities influence each other within each life cycle phase (several aspects can be represented, see sub-step one). The third step assists the user in inferring the set of project activities by reading and interpreting the previously represented relations for each life cycle phase of the project and other target entities. The resulting activities are then decomposed (using aspects selected according to sub-step one) to a level deemed suitable for the intended audience.

The first meta-methodology sub-step calls for the selection of suitable aspects (or views) to be modelled in each stage. The life cycle aspect must be present since it is essential to the meta-methodology. The selection of a MF is also recommended, as MFs typically feature structured collections of views that can be used as check-lists of candidate aspects and their intended coverage. This sub-step also calls for the identification and reconciliation of any aspect/view dependencies. The second sub-step asks the user to determine if the present (AS-IS) state of the views previously adopted needs to be shown and whether the AS-IS and future (TO-BE) states should be represented in separate or combined models. Typically, the AS-IS state needs to be modelled when it is not properly understood by the stakeholders and/or the TO-BE state is to be evolved from the AS-IS (i.e. no radical re-engineering is likely to occur). The third sub-step requires the selection of suitable modelling formalisms and modelling tools for the chosen aspects, according to the target audience of the models and also depending on the competencies and tools available in the organisation or that can be reasonably acquired.

Due to its scope and to space limitations, this chapter will cover only the first and second meta-methodology steps, focusing in particular on the benefits of creating a business model in the context of the life cycles of all relevant participant entities.

6 Application to the Environmental Management Project

Here, the meta-methodology deliverables are various models of the EM project and the EMS taking into consideration the internal and external business life cycle context. Since the management of the organisation and all other entities (business units, other organisations, agencies, laws, etc.) that need to be involved in the EM project and the EMS are to be included in the entity list (first step in Fig. 2, left), their influence will be taken into account throughout the life cycle of the EM project and the EMS. An important initial premise for EM integration into the organisation is thus fulfilled. As can be seen from Fig. 2, the meta-methodology assists in creating new knowledge (in this case, how to go about setting up and operating the EM project and the EMS) based on *context knowledge*, i.e. the know-how of running the business including corporate culture, relations with suppliers, clients, authorities, typically available at middle and top management level. The involvement of these roles in the methodology creation process establishes the conditions for management buy-in and support for the upcoming EM project and for the early involvement of the EA department in the EM project. This will create the best conditions for the integrated development of the EMS and the supporting functions of the IS.

Proposed members in the entity list are the company as a whole, business units, the EM project, the IS project, the EMS, the IS, environmental reports, NGOs, the government, EPA, EM principles (e.g. 2R, TBL), EM laws, EM standards, EM frameworks, assessment and reporting frameworks, social responsibility standards, Quality Standards and EM consultants. The MF of GERA (see Fig. 1) is adopted here as the most likely to provide a suitable formalism for the mandatory life cycle dimension and for the other selected aspects.

In this case, the TO-BE state is incremental and based on the AS-IS. Therefore, in sub-step two, it was decided that the AS-IS state must be represented for all aspects. While there is no tangible advantage in showing separate AS-IS and TO-BE states in the business model, it is very useful to do so in the decisional/organisational structure. This is because here it is imperative to clearly show where and how the functions of the EMS interact with the existing system so as to ascertain the degree of integration and effects of the EMS on the decisional and organisational structure of the host company. Separate AS-IS/TO-BE decisional/organisational models also help define several TO-BE scenarios.

A modelling formalism based on the GERA MF was chosen for the business model (see Fig. 3). GRAI-Grid [37] was selected to represent decisional and organisational aspects (see Fig. 5), together with a plain graphical editor as a modelling tool. GRAI-Grid was optimal in this case due to its high ability to represent both the decisional and the organisational aspects (note that best practice modelling principles such as formalism re-use and minimal number of languages are underlying the meta-methodology formalism selection criteria).

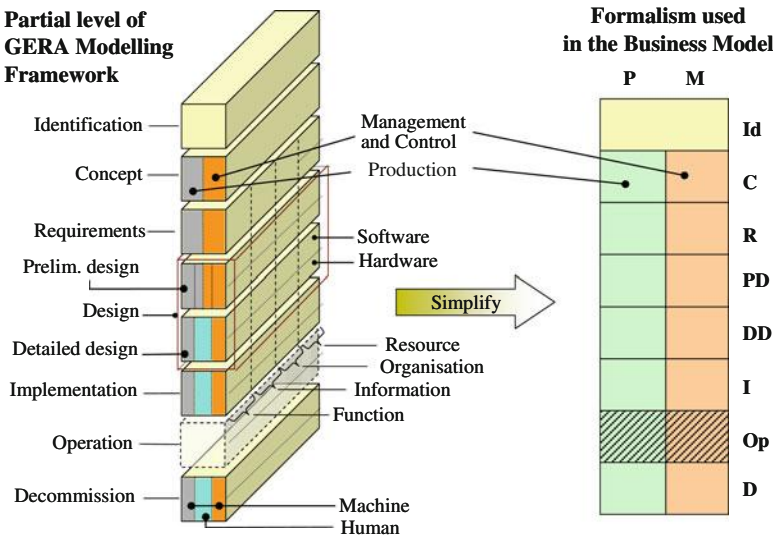


Fig. 3 Formalism used for the business model: simplified GERA MF

As shown in Fig. 2, the business model is constructed in the second step based on context knowledge (often tacit and requiring eliciting by the meta-methodology facilitator) owned by stakeholders, i.e. CxO, enterprise architect, top management, etc. A possible result is shown in Fig. 4. Here, the relations between the relevant entities can be explicitly represented for each life cycle phase. Note that some entities' life cycle representation has been reduced to the phase(s) relevant for the EM project and the EMS. For example, we are only interested in the operation life cycle phase of auditors, EM assessment/reporting frameworks, EM consultants, etc. since they are not being designed/built as part of the EM project. The figure shows the relations between the company, the EM project, the EMS and the IS, which allows building consensus, achieve a common understanding and represent what needs to be done, step by step, at a high level.

A few examples: the EMS is built by the EM project (consultants *may* also be involved in the design). The company is lobbied by NGOs and must abide by EM laws. Auditors perform certification audits (affecting the concept and design of the EMS) or surveillance audits (to check if the EMS is still compliant). The EPA will look into the EMS operation and receive information from external auditors. Importantly, the EMS should be able to redesign itself (arrow from Mgmt operation to the other life cycles) to a certain extent and thus be *agile* in the face of moderate EM regulation and market changes. Reaction to major changes should be, however, delegated to the upper company management. The arrow from the operation management side of the EMS to IS reflects the requirement to partially redesign the IS management and operation so as to integrate the EMS functions. On the other hand the IS is also influencing the design of EMS. Such

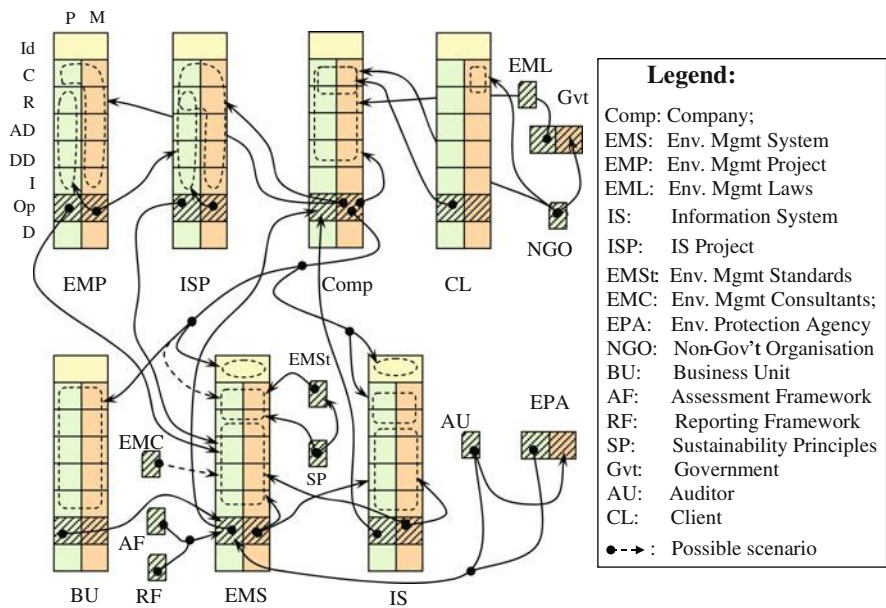


Fig. 4 Business model showing relations of relevant entities in the context of their life cycles

inter-relations are detailed in the next meta-methodology steps as controls, inputs, decision frameworks, etc.

The influences of other entities on the EMS and EM project (EMP) can also be interpreted as stakeholder concerns that translate in particular areas of interest being modelled and addressed. For example, the client may want to know how the mission and vision of the company (the concept area of Comp entity in Fig. 4) addresses its environmental concerns, and the government will want to ensure that the company abides by its environmental concerns expressed in EM laws.

Models of the AS-IS and several potential TO-BE decisional and organisational aspects have also been constructed. For example, Fig. 5 symbolically shows in a simplified form (and using the GRAI-Grid formalism) a view of the EM as an addition to the existing management that enables the organisation to manage, benchmark and improve its environmental performance in an integrated manner.

Detailed models are beyond this chapter’s scope and are available in [36].

7 Conclusions and Further Work

Currently, businesses do not seem to achieve the maximum benefits from implementing and operating an EM project and an EMS. First, there seems to be a lack of integration of the EM initiative with the business and its IS, especially at the strategic level. Thus, the management cannot take full advantage of the knowledge present in

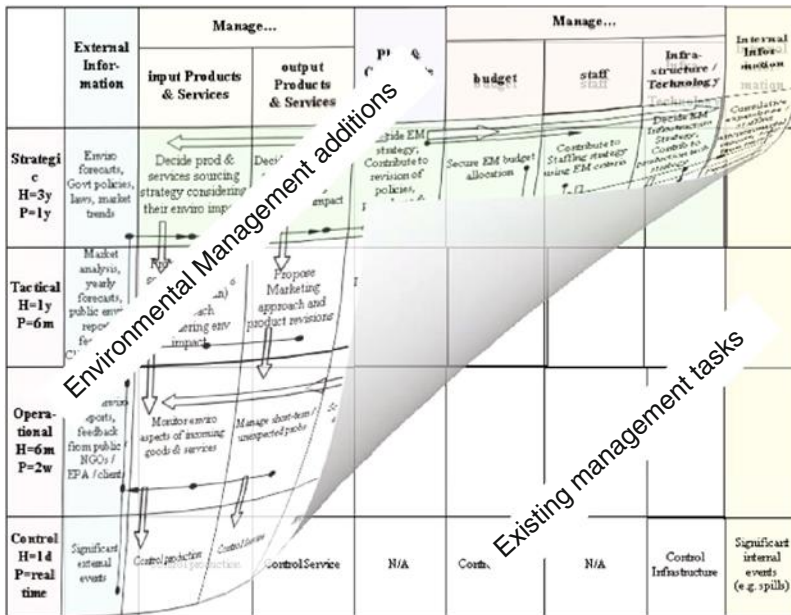


Fig. 5 EM addition to the host company management tasks

the environmental reporting mainly due to wrong format and/or level of aggregation. Second, an EMS needs to be driven internally and permeate all business areas in a consistent manner in order to produce organisational culture change, hence lasting effects. This chapter has argued that these needs are best addressed by integrating EM in the ongoing EA initiative present in some form in every successful enterprise. EA can provide the necessary artefacts and the prerequisites for a coherent, cross-departmental and culture-changing approach ensuring business sustainability and profitability in the long term.

References

1. Elkington, J. (1998) *Cannibals with Forks: The Triple Bottom Line of 21st Century Business*.
2. Blackburn, W. R. (2007) *The Sustainability Handbook*. Cornwall, UK: EarthScan Publishers.
3. UN World Commission on Environment and Development (1987) *Our Common Future (Brundtland Report)*. Oxford: Oxford University Press.
4. Molloy, I. (2007) *Environmental management systems and information management – strategic-systematical integration of green value added*, in *Information Technologies in Environmental Engineering ITEE 2007 – (Proceedings of the 3rd International ICSC Symposium)*, J.M. Gómez, et al., Editors. Springer Verlag. pp. 251–260.
5. Nilsson, I. (2001) *Integrating Environmental Management to Improve Strategic Decision-Making*. Götteborg, Sweden: Chalmers University of Technology.
6. Shewhart, W. A. (1986) *Statistical Method from the Viewpoint of Quality Control*. Dover Publications.

7. Coglianese, C., and Nash, J. (Eds.) (2001) *Regulating from the Inside: Can Environmental Management Systems Achieve Policy Goals?* Washington, DC: RFF Press.
8. ISO (2004) *ISO 14001: Environmental Management Systems – Requirements with Guidance for Use*. International Standards Organisation.
9. Willard, B. (2002) *The Sustainability Advantage: Seven Business Case Benefits of a Triple Bottom Line*. Gabriola Island: New Society Publishers.
10. Clayton, A., and Redcliffe, N. (1998) *Sustainability – A Systems Approach*. Edinburgh: Earthscan Publications, Ltd.
11. Upham, P. (2000) An assessment of the natural step theory of sustainability. *Journal of Cleaner Production* 8(6): 445–454.
12. TNEP. *The Natural Edge Project (TNEP)*. Retrieved from <http://www.naturaledgeproject.net/>.
13. Hunkeler, D. (Ed.) (2004) *Life-cycle Management*. Society of Environmental Toxicology & Chemist. Brussels.
14. EPA (2008) *Management Tools*. South Australia: Environmental Protection Agency. Retrieved Jun 2008, from <http://www.epa.sa.gov.au/tools.html>.
15. GRI (2002) *Sustainability reporting guidelines*, in *Sustainability Reporting Framework*, Global Reporting Initiative, Editor. Global Reporting Initiative.
16. ISO/IEC (2007) *ISO/IEC 42010:2007: Recommended Practice for Architecture Description of Software-Intensive Systems*.
17. Williams, T. J. (1994) The purdue enterprise reference architecture. *Computers in Industry* 24(2–3): 141–158.
18. The Open Group (2006) *The Open Group Architecture Framework (TOGAF 8.1.1 'The Book')* v8.1.1.
19. CIMOSA Association (1996) *CIMOSA – Open System Architecture for CIM, Technical Baseline, Version 3.2*. Private Publication.
20. Scheer, A.-W. (1999) *ARIS-Business Process Frameworks*. 3rd ed. Berlin: Springer.
21. DoD Architecture Framework Working Group (2007) *DoD Architecture Framework Version 1.0*. Retrieved Feb 2007, from http://www.dod.mil/cio-nii/docs/DoDAF_v1_Volume_I.pdf, http://www.dod.mil/cio-nii/docs/DoDAF_v1_Volume_II.pdf.
22. Zachman, J. A. (1987) A Framework for Information Systems Architecture. *IBM Systems Journal* 26(3): 276–292.
23. Bernus, P., Noran, O., and Riedlinger, J. (2002) *Using the Globem Reference Model for Virtual Enterprise Design in After Sales Service*, in *Global Engineering and Manufacturing in Enterprise Networks*. VTT Symposium 224, I. Karvoinen, et al., Editors. Helsinki/Finland: VTT. pp. 71–90.
24. Noran, O. (2004) *A Meta-methodology for Collaborative Networked Organisations: A Case Study and Reflections*, in *Knowledge Sharing in the Integrated Enterprise: Interoperability Strategies for the Enterprise Architect*, P. Bernus, M. Fox, and J. B. M. Goossenaerts, Editors. Toronto/Canada: Kluwer. pp. 117–130.
25. Mo, J. (2007) *The Use of GERAM for Design of a Virtual Enterprise for a Ship Maintenance Consortium*, in *Handbook of Enterprise Systems Architecture in Practice*, P. Saha, Editor. Hershey, PA: IDEA Group. pp. 351–366.
26. Noran, O. (2003) *A Mapping of Individual Architecture Frameworks (GRAI, PERA, C4ISR, CIMOSA, Zachman, ARIS) onto GERAM*, in *Handbook of Enterprise Architecture*, P. Bernus, L. Nemes, and G. Schmidt, Editors. Heidelberg: Springer. pp. 65–210.
27. Noran, O. (2003) An Analysis of the Zachman Framework for Enterprise Architecture from the GERAM Perspective. *IFAC Annual Reviews in Control. Special Edition on Enterprise Integration and Networking* (27): 163–183.
28. Noran, O. (2005) An Analytical Mapping of the C4ISR Architecture Framework onto ISO15704 Annex A (GERAM). *Computers in Industry* 56(5): 407–427.
29. Saha, P. (2007) *A Synergistic Assessment of the Federal Enterprise Architecture Framework against GERAM (ISO15704:2000 Annex A)*, in *Enterprise Systems Architecture in Practice*, P. Saha, Editor. Hershey, PA: IDEA Group. pp. 1–17.

30. Noran, O. (2007) *A Decision Support Framework for Collaborative Networks*, in *Establishing the Foundation of Collaborative Networks (Proceedings of the 8th IFIP Working Conference on Virtual Enterprises – PROVE 07)*, L. Camarinha-Matos, et al., Editors. Guimaraes/Portugal: Kluwer Academic Publishers. pp. 83–90.
31. ISO (2000) *Annex C: GERAM*, in *ISO/DIS 15704: Industrial Automation Systems – Requirements for Enterprise-reference Architectures and Methodologies*. International Standards Organisation.
32. Noran, O. (2004) *A Meta-methodology for Collaborative Networked Organisations: A Case Study and Reflections*, in *Knowledge Sharing in the Integrated Enterprise: Interoperability Strategies for the Enterprise Architect*, P. Bernus, M. Fox, & J.B.M. Goossenaerts, Editors. Toronto: Kluwer. pp. 117–130.
33. Noran, O. (2005) *Managing the Collaborative Networks Lifecycle: A Meta-methodology*, in *Advances in Information Systems Development – Bridging the Gap between Academia and Industry (Proceedings of the 14th International Conference on Information Systems Development (ISD 2005))*, A. G. Nilsson, et al., Editors. Karlstad, Sweden: Kluwer. pp. 289–300.
34. Noran, O. (2006) Refining a meta-methodology for collaborative networked organisations: A case study. *International Journal of Networking and Virtual Organisations* 3(4): 359–377.
35. Noran, O. (2007) *Discovering and Modelling Enterprise Engineering Project Processes*, in *Enterprise Systems Architecture in Practice*, P. Saha, Editor. 2007, Hershey, PA: IDEA Group. pp. 39–61.
36. Noran, O. (2008) *A Meta-methodology for Collaborative Networked Organisations: Creating Directly Applicable Methods for Enterprise Engineering Projects*. Saarbrücken: VDM Verlag.
37. Doumeingts, G., Vallespir, B., and Chen, D. (1998) *GRAI Grid Decisional Modelling*, in *Handbook on Architectures of Information Systems*, P. Bernus, K. Mertins, and G. Schmidt, Editors. Heidelberg: Springer. pp. 313–339.

Effective Monitoring and Control of Outsourced Software Development Projects

Laura Ponisio and Peter Vrugink

Abstract In our study of four outsourcing projects we discover mechanisms to support managerial decision making during software development processes. We report on Customer Office, a framework used in practice that facilitates reasoning about projects by highlighting information paths and making co-ordination issues explicit. The results suggest a key role of modularisation and standardisation to assist in value creation, by facilitating information flow and keeping the overview of the project. The practical implications of our findings are guidelines for managing outsourcing projects such as to have a modularised view of the project based on knowledge domains and to standardise co-ordination operations.

Keywords Software development · Outsourcing · Project management

1 Introduction

Software development involves both managerial and technical decisions. Extensive research has been done on concepts and best practices to improve project management, such as software cost estimation models (e.g. COCOMO), development process frameworks (e.g. RUP), maturity models (e.g. CMMI), governance models (e.g. Cobit), and project management methods (e.g. Prince2).

However, in practice, project managers drive projects based mostly on experience. They still find it difficult to know which mechanisms are useful to control large software development projects. Specifically, practitioners ask for effective mechanisms to control software projects and to connect strategic, technical and organisational domains.

Outsourcing software development presents extra challenges because development is performed in an inter-organisational context. A customer (an organisation) asks a vendor (another organisation) to produce some IT artefact, such as a

L. Ponisio (✉)
University of Twente, Twente, The Netherlands
e-mail: ml@ponisio.com

software application. Although organisations collaborate transferring knowledge from customer to vendor (e.g. requirements) and from vendor to customer (e.g. appropriate technology), each has its own interests and needs, which often conflict. Tacit requirements, conflicting interests and knowledge-domain gaps add up to generate final solutions that cost more in terms of resources than what was originally planned or that do not help customers to meet their ambitions.

The objective of this work is to understand mechanisms that organisations put in place to optimise software development management in outsourcing projects. In particular, this work aims to gain knowledge about which mechanisms facilitate transfer of the information that managers need when making decisions during software development.

We have studied four projects representative of software development in an outsourcing context. The results suggest a key role of modularisation and standardisation as effective mechanisms to manage development projects towards value creation.

Customer Office (CO), a reasoning framework found in practice in the observed organisation, serves as an example to explain this modularity observed in practice. CO has the potential to be an effective operationalisation of CMMI for acquisition module because it highlights information paths and makes co-ordination issues explicit. Prioritising reasoning about the project in terms of working units, CO encourages project managers to keep the oversight of co-ordination. Putting co-ordination up-front, project managers have better control of the information flow, which is vital to transfer requirements effectively and to gather technical information necessary to make informed decisions.

The practical implications of our findings are guidelines that include having a modularised view of the project according to knowledge domain and standardising co-ordination operations. At the organisational level, results point at the importance of having managers with *experience* and of providing managers and engineers with managerial information that managers can relate to.

2 Maximising Value Creation in Software Development

To maximise value creation, project managers must understand the connections between technical decisions and enterprise-level value. With inadequately understood connections, project managers are unable to make decisions that could significantly increase the value created by software development. Consider software modularity. The ability to meet time-to-market requirements depends on having a modular design. An independent-feature-based architectural style helps developers to meet time-to-market requirements because it enables them to abandon unimportant features later if times runs out. Project managers in software development strive, thus, to connect technical decisions with value creation criteria. Practitioners ask for mechanisms that help them to connect strategic, technical and organisational domains. In other words, mechanisms that help them to answer the question “how are we in control?”

2.1 *Related Work and Existing Solutions*

Solving the technical-value mismatch occurs in the context of two domains: software development (in the area of software engineering) and management in IT outsourcing projects (in the area of IT project management). Decision making is the linking pin between these domains because better decision making is a key enabler of business–IT alignment. The following sections elaborate on the related work in these areas.

2.1.1 Support for Decision Making in Software Development

One way to exercise control is to connect technical properties of the software product with decisions supporting value maximisation. Managers do this by measuring properties of the software (to follow closely the development needs), by estimating the cost (or effort) required to develop a system and by choosing good models for software development. The next paragraphs elaborate on these issues.

Software Metrics

This term describes a range of activities related to measurement in software engineering. These activities include measuring some property of the software code (such as size and complexity) and quantitative aspects of quality control and assurance [5].

One of the major reasons for using software metrics is to improve the way decision makers monitor, control and predict various attributes of software and the software development process. Furthermore, metrics are also used to measure software product quality and evidence of use of metrics is needed to achieve higher levels of the Capability Maturity Model (CMM).

The major problems of metrics are using measurements in isolation, handling uncertainty and combining them with evidence. After 40 years of research, we have learnt that as important as the metric is its *application*. Metrics must, thus, have clear goals and objectives.

Software Cost Estimation Models.

Software cost estimation is concerned with making estimates of the effort required to complete the software for a system development project [15]. Research in the area of software engineering and economics has produced a number of software cost estimation models such as COCOMO, PRICE S, SLIM, SEER, SPQR/Checkpoint and Estimacs. They are still in use today. A recent systematic review identified 304 software cost estimation papers in 76 journals [7].

In spite of much theoretical support to reason about software cost, support to reasoning about benefits and value is sub-optimal. The models need to adapt to support reasoning about benefits and cost in a context that rapidly shifts priorities. For instance, estimation models need to be integrated to the current

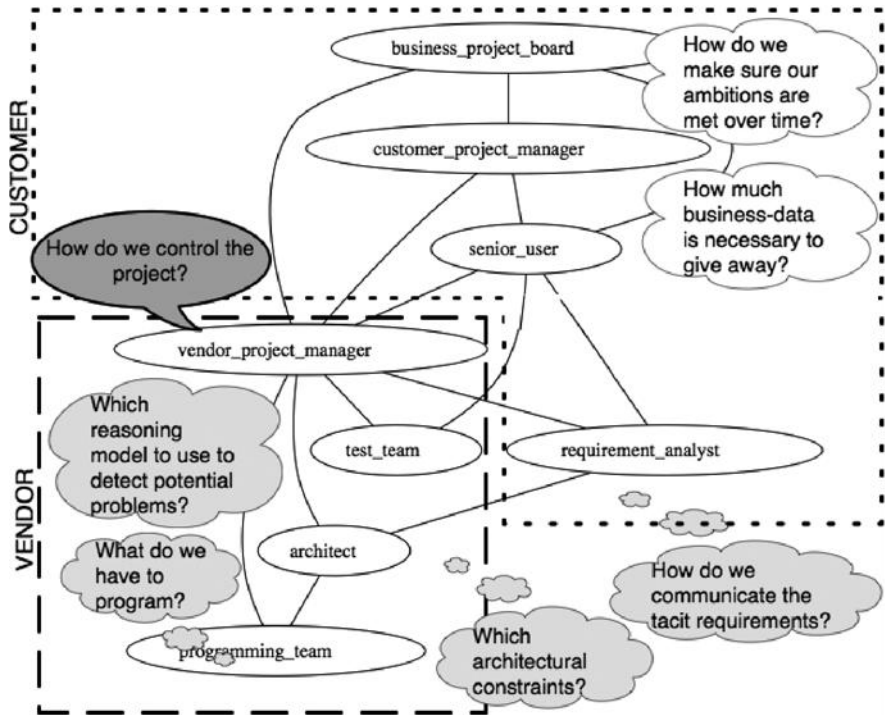


Fig. 1 Outsourcing actors forming an inter-organisational network to carry on an IT solution-development project

inter-organisational way of developing software. Outsourcing introduces *multiple stakeholders* whose conflicting interests need to be addressed. Outsourcing software development by definition involves co-operation among several organisations (see Fig. 1). Development in an inter-organisational context challenges the organisation's operating model – the necessary level of business *process integration* and *standardisation* for delivering goods and services to customers.

Software Development Models

They correspond to methods used to support software development activities. Several solutions from perspectives ranging from requirement techniques to agile development have been proposed, for example, V-Model, Spiral model, Waterfall model, RUP, Iterative model, Agile, Scrum and eXtreme programming. All these methods and techniques have advantages and disadvantages. Success in their application depends on the project characteristics and the way they are applied. Unfortunately, managers have very few guidelines regarding ways to operationalise them.

All in all, we observe that results from software engineering are not presented in terms that clearly show value for controlling software development. It is understandable that practitioners express the need for more practical mechanisms.

2.1.2 Management in IT Outsourcing Projects

Scholars try to understand governance from two main different perspectives: connecting business and technology (from a business Information Technology (IT) alignment perspective) and managing the project (from a governance perspective). In both perspectives, outsourcing introduces new socialisation and co-ordination issues related to the customer–vendor relationship.

Business IT Alignment.

Establishing the connection between technology objectives and business objectives impacts on the value of the software development results [10]. The field is very active. A recent literature review on IT alignment [3] describes over 150 alignment articles. Research highlights the importance to share responsibility for alignment, build the right culture, educate and equip, share knowledge, and manage the IT budget. It has been pointed out that needs for objective alignment differ between organisations [13] and available governance options have been described [11].

However, little research investigates how, when and where we can improve IT alignment. In fact, according to Gutierrez et al., “the variety of approaches proposed has created confusion about the applicability and context in which these approaches can be used” [6]. That confusion explains why, in practice, managers rely mostly on their experience to choose the adequate theoretical constructors and their practical use in a specific project.

Governance in IT Outsourcing.

According to Weill there are five areas where IT decisions have to be made (IT principles, IT architecture, IT infrastructure strategies, business application needs and IT investment and prioritisation) and there are several styles. Top-performing firms use particular combinations [16]. Many concepts and best practices have been proposed (e.g. ASL, BiSL, DSDM, RUP, CMMI, CobiT, Prince2 and ITIL). While existing research has produced useful models and propositions, there is need for empirical research that bridges the gap between software development and governance.

All in all, we can observe that many concepts and best practices have been proposed. Yet, a major problem remains: it is unclear which mechanism to use. In spite of significant research efforts, there is need for practical findings that help practitioners to know when, how and where can they exercise control in outsourced software development projects.

3 Research Method

We carried out a case study. Drawing from established processes of investigation aimed at the discovery of facts [4], case study research is appropriate to investigate the how and why of projects occurring in reality. Klein and Meyers demonstrate that case study research can be interpretive and indicate seven principles of interpretive research [9], which we followed in our case study.

According to Klein and Meyers, interpretive research attempts to understand phenomena by gaining knowledge of reality through social constructions, documents, tools and other artefacts. Interpretive research focusses on the complexity of human sense making as the situation emerges rather than predefining dependent and independent variables. It is appropriate for our research because our problem consists of gaining knowledge. It helps us, thus, to understand phenomena by interpreting the software development situation as it emerged in practice.

3.1 Research Design

We are interested in finding how managers deal with software development issues such as complexity and requirements transfer across domains. First, we ponder how the elements controlling offshore outsourcing software development interact in real-life projects. For instance, we consider which artefacts (such as Requirements Management Plan, Software Architecture Document, Master Test Plan, Use Cases and Financial Status Report) were mandatory for the delivery of standard offshore, outsourcing development projects and which artefacts were essential in practice in such projects.

Second, we studied documentation of such projects and a best practices framework used in the organisation under study. In doing so, we focussed on finding actors, artefacts, team organisation, information exchange and dependencies. We paid special attention to recurrent problems (such as tacit requirements) and counter-acting measures. To that regard we consulted experts so we could learn their opinions and tips. We observed four large offshore outsourcing projects and analysed any mechanism we could observe.

Finally, we focussed on any mechanism that seemed novel (against the study of the state of the art) and potentially effective to control these projects. Since it was an iterative study, observation and analysis made us gain knowledge of reality, understanding the phenomena. Eventually, all the details acknowledged were in harmony and the moment of understanding had arrived for us. To prevent misunderstandings and threat to internal validity, once we had our findings, we confirmed them by way of interviewing expert project managers (see Section 5.2).

3.2 Data Collection Method

Data was collected using a combination of methods: interviews, unstructured observations of documentation, and focus groups. According to Eisenhardt [4], collecting

data from different types of sources gives more strength to validity. Thus, the evidence was gathered from different sources, where evidence found in one source (e.g. interviews) corroborates evidence found in another (e.g. observations and focus groups).

The observations took place between January 2008 and March 2009 and were made by the two researchers. To double check the findings and to detect potential misunderstandings, focus groups and extra interviews with three experts were performed. The experts were software architects each with more than 10 years of experience in managing software development projects. A semi-structured interview protocol was used to allow the participants to clarify terms and to investigate issues that could improve the description of the situation. Participants were guaranteed anonymity and information was sanitised.

3.3 Case Selection

Our case study is based on four projects executed in an organisation that we name Big. Being a multinational IT service provider, Big has offices in many countries. Big is a good example of information technology (IT) service provider because it exemplifies most (large) offshore outsourcing development organisations. In The Netherlands, with several thousands of employees, Big is in the top-ten of IT service providers ranked to number of employees and revenue.

The four projects that we observed in Big reproduce the state and behaviour of traditional outsourcing projects, and it is reasonable to consider these projects representative of their kind. These projects are examples of offshore in-house development of software on behalf of Big's customers carried out by geographically distributed teams formed by 10–46 members.

4 Case Study

The projects observed develop information technology (IT) solutions in the context of outsourcing. As such, they evolve around the relation between a customer, a domestic (on-site) team and an offshore team. Both teams belong to Big. The difficulties that managers of Big experience are related to problems to control large complex projects with geographically (and culturally) distributed teams. In particular, challenges are related to large project size and complexity, and requirements transfer (e.g. sub-optimal understanding of tacit requirements or too late understanding).

Since these projects are large, customer and vendor need to put in practice mechanisms to (a) manage the complexity of designing in the large and to (b) maintain the consistency between business goals and the system's architecture in spite of having multiple teams geographically distributed. Big's strategy with these projects is to be customer oriented and to divide and conquer. This strategy manages their strategic goal to build long-term partnerships with the customer. In practice, this

means to have part of the staff working closely with the customer and to govern the project emphasising the client’s view. In the next sections we explain the practices we observed.

4.1 Customer Office

Recently, project managers implemented mechanisms to optimise management of software development in outsourcing projects. Among those mechanisms we find a reasoning framework called Customer Office (CO). This framework was developed bottom-up by the second author of this chapter and his team.

CO is a reasoning framework based on Result Delivery (RD), a model inspired by Prince2 [1,2]. RD is organised around key process areas of CMMI acquisition. Figure 2 depicts the RD model. Practitioners implemented the CMMI acquisition module and operationalised it through CO. In the documentation, we observed that RD encourages organising the working force into *units* (e.g. development centre, business project board, IT project board) organised by *knowledge domain*.

The CO framework places the IT project team in the centre of co-ordination activities, helping managers to derive IT solutions that meet the customer’s business goals by improving the link between the Customer and the Development Centre.

CO emphasises collaboration issues with the client (who has the business know-how) and smoothes the way for us to focus on value added to the overall business process. **The client is always responsible for the project.** [Emphasis added by the interviewee.]

As Fig. 2 shows, this organisation minimises coupling between units concerned with different domains (such as managers and developers), maximising cohesion

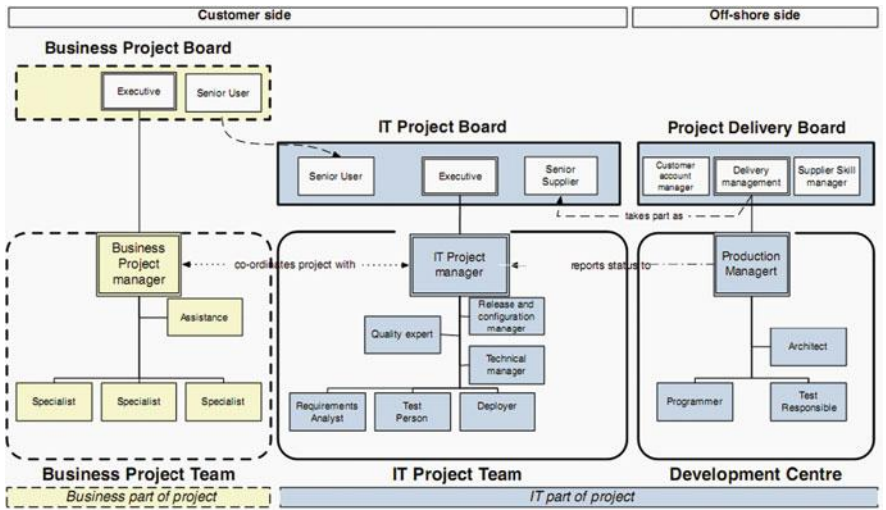


Fig. 2 The result delivery (RD) model

among units (e.g. a development unit was devoted to programming one package). Coupling and cohesion are attributes that describe communication across units and the relationships within a unit, respectively.

4.2 Experience

The main challenges of the observed projects are their size and complexity. Geographical and domain knowledge distance hinder effective requirements transfer across business and technical domains. Moreover, size makes it difficult to recognise essential data and to keep the overview of activities. Recognising, for example, which of the 72 encouraged documents is essential to detect potential problems and opportunities to increase value is performed mostly on the basis of the manager's *experience*.

We know that we have to make sure the following reports are made [thoroughly]: Financial Status Report, Quality Status Report, Project Status Report, Customer Status Report and summary of both.

These are the reports managers find most important. One explanation is that managers can relate to these documents because their contents have meaning for them. These reports provide data that managers use to discuss with the customer. In particular, the domain knowledge related to these documents corresponds to the domain knowledge of the IT Project Board unit, which is close to the domain knowledge of the Business Project Board unit.

We observed that in general project managers have years of experience. Of course, generating experience demands time. In Big, employees are encouraged and supported to assist to training that is expected to benefit both their job and long-term career plans.

5 Results

The results of our study show that various mechanisms were used in those projects. Means through which the observed organisation optimises management of software development projects are modularisation of activities and standardisation of co-ordination mechanisms.

Modularisation facilitates control over co-ordination across units because it prevents jeopardising the connections between technical issues and business goals (which hinder value creation if badly managed).

The realisation of this element in the project corresponds to the unit structuring, specifically, as observed in RD and operationalised in CO. Viewing the project in terms of RD's units facilitates *reasoning about the project* because it highlights the information paths and it makes explicit co-ordination issues. In other words, prioritising reasoning about the project in terms of these working units, RD encourages project managers to pay attention to co-ordination. Putting co-ordination up-front,

project managers have better control on the information flow, which is vital to the effective transfer of requirements and to make informed decisions. We, thus, interpret CO as an effective mechanism to increase co-ordination among the units, which is essential to deliver software products with value.

In addition, results highlight the importance of supporting experience by providing managers with managerial information that they can relate to, i.e. information that has meaning to them because it is at least partially within their knowledge domain. In our study, we observed that among the 72 suggested documents, only five were key (according to the project managers).

We learn that in Big, transferring technical information to managers works well because they have found mechanisms to co-ordinate effectively all the project units: cross-team communication is enhanced by a modularised organisation, favouring co-ordination, and by standardised co-ordination practices, favouring predictability and repeatability in translating across knowledge domains, languages and cultural factors. The practices observed match common challenges of geographically distributed teams and are in line with their CO reasoning framework.

5.1 Practical Implications

5.1.1 Have a Modularised View of the Project

A modularised organisation happens when there are standardised procedures to communicate information across well-defined units. Actors within a unit share knowledge domains. Links across units are few and explicit. RD makes units and their links explicit.

In our observations, modularisation was implemented by structuring activities into the diverse units (e.g. development centre, business project board, IT project board). We interpret that in Big, modularisation was effectively used to find ways to maximal value throughout the decision space. Managers make better decisions because they have the information they need: there are explicit roads to share cross-domain information.

One way to implement modularisation of units is to follow the RD model. We interpret that RD responds to the manager's needs to control customer-oriented projects. Specifically, it can be an effective view to reason about projects where customer and vendor are seeking long-term partnership. Under these circumstances, the RD view helps managers to maintain consistency between customer's business goals and IT solution because it facilitates communication between different groups of stakeholders. This scenario, in turn has the potential to facilitate quality requirement transfer and to support informed decision making.

5.1.2 Standardise Co-ordination Operations

Standardisation of operations happens when an organisation puts in place standards (e.g. mandatory documents with explicit communication protocols) intended

to improve predictability and repeatability. The realisation of this element in our case study corresponds to their reasoning framework: CO is organised around key process areas of CMMI acquisition module. For instance, every unit has a responsible person and everybody (also people from other units) knows who has been appointed to this task. Therefore, every team member knows “whom to call”.

The rationale behind standardisation is that it facilitates communication because the processes in place are made explicit (e.g. who is responsible for what). Co-ordination, speed to market and flexibility to change are also improved. The flexibility obtained is paramount in a context signalled by inter-organisational development.

5.2 Validation

5.2.1 External Validity

External validity (can it be generalised to other cases?) is something we cannot prove, but the results of this study are encouraging at the very least. Our findings are based on the study of only four projects and therefore the criterion of transferability is limited. Nevertheless, we believe the results of our study could be generalised for the following reasons: first, the organisation and the projects observed are representative of software development outsourcing projects, second, because our results are in line with existing research [8, 12, 14], and third, because the outsourcing experts we interviewed can relate to them.

5.2.2 Internal Validity

Internal validity (do our interpretations lead to the right conclusions in this case study?) has been established as high by experts. According to them, our interpretations led to the right conclusions, which means that the internal validity criterion is met. The study approach deserves some comments. Klein and Meyers [9] have suggested a set of principles for the conduct and evaluation of interpretive field research in information systems. The research presented in this chapter followed Klein and Meyer’s principles.

6 Conclusion

This chapter reports on a study of four outsourced software development projects. The objective of the study is to learn the mechanisms that organisations put in place to control software development projects in the context of outsourcing.

The results show that various mechanisms were used in those projects. Means through which the observed organisation optimises management of software development projects are modularisation of activities and standardisation of co-ordination mechanisms. Project managers reason about the project in terms of units organised

by knowledge domains. This organisation has the advantage to make co-ordination issues explicit, improving keeping the overview of the project.

A second contribution of this chapter is reporting on CO, a reasoning framework found in practice during our observations. CO was developed bottom-up by the second author and his team and supports a modularised view of the project. In it, practitioners implemented and operationalised the CMMI acquisition module. Making co-ordination issues such as information paths explicit, CO empowers project managers with a reasoning tool to better control transfer of requirements, and to gather technical information necessary to make informed decisions.

We believe CO has the potential to facilitate management, supporting modularisation and standardisation in the multiple domains of outsourcing projects, which helps managers to keep the project's overview. Future work will determine if CO effectively supports dynamic monitoring and control of complex software development activities.

Acknowledgements This work was supported by The Netherlands Organisation for Scientific Research, project nr. 638.004.609 (QuadRead).

References

1. Office of Government Commerce. (2005) *Managing Successful Projects with PRINCE2*. 4th ed. The Stationery Office. PRINCE2 Manual. Great Britain.
2. Office of Government Commerce. (2005). *Projects in Controlled Environments (Prince)*. Office of Government Commerce. (OGC). Accessed March 13, 2009. http://www.ogc.gov.uk/methods_prince_2.asp.
3. Chan, Y. E., and Reich, B. H. (2007) IT alignment: An annotated bibliography. *Journal of Information Technology* 22: 316–396.
4. Eisenhardt, K. M. (1989). Building theories from case study research. *Academy of Management Review* 14(4): 523–550.
5. Fenton, N. E., and Neil, M. (2000). Software metrics: Roadmap. In *ICSE '00: Proceedings of the Conference on The Future of Software Engineering*. New York, NY: ACM. pp. 357–370.
6. Gutierrez, A., Orozco, J., and Serrano, A. (2008) Developing a taxonomy for the understanding of business and IT alignment paradigms and tools. In *Proceedings of the Sixteenth European Conference on Information Systems*, T. Acton, J. Conboy, and W. Golden, Editors. Galway: National University of Ireland, Galway (in print).
7. Jørgensen, M., and Shepperd, M. (2007) A systematic review of software development cost estimation studies. *IEEE Transactions on Software Engineering* 33(1): 33–53.
8. Kernkamp, R. (2007) *Alignment of Requirements & Architectural Design in a Blended Delivery Model*. Master's thesis, University of Twente.
9. Klein, H. K., and Meyers, M. D. (1999) A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS Quarterly* 23(1): 67–93.
10. Luftman, J. (2000) Assessing business-IT alignment maturity. *Communications of AIS* 4: 1–50.
11. Miranda, S. M., and Kavan, C. B. (2005). Moments of governance in IS outsourcing: Conceptualizing effects of contracts on value capture and creation. *Journal of Information Technology* 20(3): 152–169.
12. Ponisio, M. L., and Vrugink, P. (2008). Analysing boundary objects to develop results that support business goals. *2008 International Conferences on Computational Intelligence for Modelling, Control and Automation; Intelligent Agents, Web Technologies and Internet*

Commerce; and Innovation in Software Engineering. Los Alamitos, CA: IEEE Computer Society. pp. 516–521.

13. Reich, B. H., and Benbasat, I. (1996) Measuring the linkage between business and information technology objectives. *MIS Quarterly* 20(1): 55–81.
14. Ross, J. W., and Westerman, G. (2004) Preparing for utility computing: The role of IT architecture and relationship management. *IBM Systems Journal* 43(1): 5–19.
15. Walkerden, F., and Jeffery, D. R. (1997) Software cost estimation: A review of models, process, and practice. *Advances in Computers* 44: 59–125.
16. Weill, P. (2004) Don't just lead, govern: How top-performing firms govern IT. *MIS Quarterly Executive* 3(1): 1–17.

Classification of Software Projects' Complexity

P. Fitsilis, A. Kameas, and L. Anthopoulos

Abstract Software project complexity is a subject that has not received detailed attention. The purpose of this chapter is to present a systematic way for studying and modeling software project complexity. The proposed model is based on the widely known and accepted Project Management Body of Knowledge and it uses a typology for modeling complexity based on complexity of faith, fact, and interaction.

Keywords Software project management · Software project complexity

1 Introduction

Software projects are complex endeavors and in many cases their outcome is far from being certain. This has been proven by many studies and on various project types. As a result, the track record of software engineering industry is rather disappointing [1, 2]. This implies, at least, that software projects are complex undertakings.

Traditionally, complexity in software projects is measured implicitly: either by measuring the software project product or by measuring characteristics of the software process. A large number of metrics have been described in the literature for measuring different characteristics of software and software projects, such as size, complexity, reliability, and quality [3]. Respectively, for software processes five are the perspectives that are central to measurement: performance, stability, compliance, capability, and improvement [4].

These approaches in measuring complexity are not sufficient since “in complex systems the whole is more than the sum of parts” and “given the properties of the parts and the laws of interaction, it is not trivial to infer the properties of the whole” [5]. This directs us to the observations that complexity can only be measured holistically.

P. Fitsilis (✉)

Technological Education Institute of Larissa, 41110, Larissa, Greece
e-mail: fitsilis@teilar.gr

In this chapter, we present a holistic model for measuring the complexity of software projects. The model is built by combining Project Management Body of Knowledge (PMBOK) [6] with Patterns of Complexity [7]. PMBOK defines nine project management knowledge areas which are Project Integration Management, Project Scope Management, Project Time Management, Project Cost Management, Project Quality Management, Project Human Resource Management, Project Communications Management, Project Risk Management, and Project Procurement Management. PMBOK exhibits a number of characteristics that make its usage attractive for measuring the software project complexity: it is well known, it combines knowledge required with necessary processes, and it is analytical.

Despite the advantages, PMBOK is not sufficient for measuring project complexity since complexity appears in many different forms. According to Geraldi's [7] typology of complexity, "complexity of faith" is due to uncertainty, "complexity of fact" refers to the amount of interdependent information, and "complexity of interaction" is present in interfaces between systems or locations of complexity.

The model is built by defining for each PMBOK knowledge area and for each complexity category a set of metrics which allows the measurement of project complexity in a robust and multifacet manner that takes into account project management knowledge, processes, and patterns of complexity.

2 Project Complexity and Patterns of Complexity

Project complexity is a common concept recognized in a number of different ways. It is given a number of different interpretations based on the reference context or on each individual's experience. In many cases project complexity is used as a replacement for project size, or alternatively to project difficulty. Further, in other cases project complexity is perplexed with project's product complexity. Definitely there is a lack of consensus on what project complexity really is [8, 21].

The dictionary definition of complex is something involving of different but related parts, or difficult to understand [9].

2.1 Complexity and Size

Project size is an important characteristic that to a large extent can capture the complexity of a project. Size is one of the most important attributes of software and, in most cases, it is directly related to the effort required, the productivity of the team, the required quality, etc.

Commonly used software sizing methodologies are counting the Lines Of Code (LOC) of the source code [10], Function Point Analysis (FPA) [11], Use Case Points (UCP) [12], COCOMO II [13], etc.

Even though complexity is present as one of the systems' characteristics examined, for estimating the size, in most of the above methodologies, e.g., "Complex

Processing” in FPA and “Complex Internal Processing” in UCP, project size alone cannot be used for measuring the complexity of a project. This applies since a large but well-structured software project with relaxed cost and time constraints can be much less complex in comparison with a relatively small-in-size project which has a highly integrated product design and limited budget and/or time-to-market objectives.

2.2 Product Versus Project Complexity

The terms “product complexity” and “project complexity” in many cases are used interchangeably [14]. Baccarini [15] regards product complexity as a subcategory of technological complexity, which covers complexities related to products and processes. Usually, when we are measuring the complexity of the software product, our objective is to predict and to understand what causes increased defects and lower productivity. Therefore, more development effort is required and as a result the software project is more complex.

Software complexity is demonstrated in three different forms: structural, conceptual, and computational. Structural complexity looks at the design and structure of the software itself; conceptual complexity refers to difficulty in understanding the system/requirements and/or code itself and the novelty/newness of the software product; and finally computational complexity refers to the complexity of the computation being performed [3, 16].

2.3 Organizational and Technological Complexity

Baccarini [15] defined two facets of project complexity: the organizational facet and the technological facet.

Organizational complexity is defined as the amount of differentiation that exists within different elements constituting the organization [17, 22]. The differentiation has two dimensions: vertical (depth of organization) or horizontal (organizational structure, task structure). One important characteristic of organizational complexity especially in projects is the ability or inability of organizational elements to connect and to interact.

The second facet of complexity, as defined by Baccarini, is technical complexity. More specifically, he defines technological complexity by differentiation, which refers to the variety or diversity of some aspect of a task, and technological complexity by interdependency, between tasks, within a network of tasks, between teams, etc.

2.4 Patterns of Complexity

Geraldi [7, 18, 19] and Williams [20] defined three types of complexity:

Complexity of Faith (CoFaith): It is synonymous to uncertainty and it is present since projects are creating something unique or solving new problems. Usually, at the beginning of a project, we are dealing with uncertainty (on resources, effort required, etc). However, the uncertainty is decreasing as the project progresses and as a result CoFaith is transformed to Complexity of Fact. CoFaith cannot be measured objectively. Even when there is a quantitative way to measure CoFaith this is based on subjective opinions.

Complexity of Fact (CoFact): It is similar to structural complexity and considers complexity as an intrinsic property of the software system. The most important problem in measuring the CoFact is that it relates to a very large number of interdependent information. An attempt to exhaustively analyze and measure all the contributing factors will fail and therefore we should always keep a holistic view on the project complexity measurement.

Complexity of Interaction (CoInteraction): It is present in interfaces between systems, locations, and humans and it is characterized by transparency, multiplicity of reference, and empathy.

3 Project Management Body of Knowledge

As it was mentioned before, Project Management Body of Knowledge (PMBOK) [6] is defined in terms of process groups and knowledge areas. In this study, we will focus on the knowledge areas, since these areas are offering a more precise idea of what project management is about and at the same time they give the overall picture on how to measure project complexity. The knowledge areas defined in PMBOK are the following:

Project Integration Management describes the processes and activities needed to identify, define, combine, unify, and coordinate processes and project management activities. The main objective during the initial phase of the project is to define the project charter and to develop the project plan, while at later phases the emphasis is shifted to monitoring and controlling the project plan. Obviously, this initial phase exhibits increased uncertainty and is inherently complex. A large part of the project integration management subject area is the change management. Change management is the process of requesting, determining attainability, planning, implementing, and evaluation of changes to a software system. Large number of changes implies increase in system volatility and complexity (i.e., through changes the structure of a system becomes more complex).

Project Scope Management encapsulates processes for ensuring that project work is defined, verified, and controlled. Due to the nature of software projects, where the project product is intangible, scope management presents increased difficulty and it adds complexity to the project. Therefore, the problem of managing software requirements is listed as one of the most difficult in the software life cycle.

Project Time Management describes the processes concerning the timely completion of the project. It includes the definition and sequencing of the activities,

estimation of their duration, estimation of the resources needed, and development of project schedule. Factors such as number of activities, difficulty of activities, number of resources involved, tidiness of schedule, etc. are all influencing the complexity of the project.

Project Cost Management includes processes involved in estimating budgeting and controlling cost.

Project Quality Management describes the processes involved in assuring that the project will satisfy the objectives for which it was undertaken. Quality objectives set during the quality planning phases can introduce complexities to software projects.

Project Human Resource Management includes all necessary processes for organizing, managing, and leading the project team. Organization complexity is defined in the literature as one of the most important sources of project complexity. Further, factors such as team size and number of different professions required can contribute significantly to project complexity.

Project Communications Management describes the processes concerning the communication mechanisms of the project and relates to the timely and appropriate generation, collection, dissemination, storage, and ultimate disposition of project information. It is related directly to organizational complexity and it is affected by the number of project stakeholders.

Project Risk Management describes the processes concerned with project-related risk management. Risk is a future, uncertain event that if it occurs has a negative effect on scope, time, cost, or quality. Therefore, the complexity in risk management is related to the process for handling the risk rather than to the risk itself.

Project Procurement Management includes all processes that deal with acquiring products and services needed from third parties for completing the project. The number, size, and novelty of the procured goods or services can influence the complexity of the project.

4 Sources of Complexity

The first step for developing a model for systematically measuring complexity is defining a set of software project characteristics that are the sources of complexity. An exhaustive list with sources of complexity would be enormously large, and at the end it is of no practical use since it would be proven difficult to measure everything. Further, it would be impossible to agree on the exact scope of each complexity source or on the definition. Literature provides significant evidence on the validity of the above arguments since numerous authors are proposing different typologies, characteristics, attributes, etc.

Therefore, it is necessary to use a well-established and widely accepted framework for classifying the sources of complexity and subsequently for studying project complexity. This can be achieved by combining a project management framework, such as PMBOK, with a complexity typology, such as “patterns of complexity.”

Table 1 presents a summary of the identified source of complexity along with indicative metrics that could be used.

Table 1 Taxonomy of complexity

PMBOK subject area	Sources of complexity (type of complexity)	Indicative metrics
Project Integration Management	The complexity of the planning process (CoFact)	Number of project management processes employed
	Executing organization immaturity (CoFact)	Level of maturity (e.g., CMM level)
	Software requirements volatility (CoFact)	Number of change requests
	Clarity of project objectives and management commitment (CoFaith)	Number of stated objectives Financial resources committed
	Software development methodology (CoFaith)	Number of phases Level of rigidity/agility
	Project novelty (CoFaith)	Number of similar software projects within organization Number of similar software projects in the literature
	Project environment (CoFaith)	Legislation and regulations Market and competition
Project Scope Management	Software size (CoFact)	Metrics such as FPA and UCP Number of deliverables
	Software structure and architecture (CoFact)	WBS levels Number of components and modules Number of third-party components used Number of different technologies used within project
Project Time Management	Project schedule (CoFact)	Project duration Number of activities Number of organizations (units) involved Number of different resources needed
	Project schedule difficulty (CoFact)	Number of dependencies between activities Number of activity constraints
Project Cost Management	Project budget structure (CoFact)	Budget size Number of different budget lines Level of accuracy
Project Quality Management	Quality planning (CoFact)	Number of quality metrics employed Number of quality reviews

Table 1 (continued)

PMBOK subject area	Sources of complexity (type of complexity)	Indicative metrics
Project Human Resource Management	Size and structure of the team (CoFact)	Team size Variety of different skills needed Personnel availability and mobility Geographical distribution
	Personnel expertise and quality (CoFaith)	Personnel technological experience Personnel problem domain experience Personnel process familiarity
Project Communication Management	Reporting requirements (CoFact)	Reporting frequency Number of different reports
	Information flow (CoInteraction)	Different number of communication flows Number of planned meeting
Project Risk Management	Actors involved in the project (CoInteraction)	Number of different stakeholders
	Risk identification, quantification, and response planning (CoFact)	Number of risks identified Number of risks quantitatively analyzed Number of risks where a mitigation plan is available
Project Procurement Management	Procurement planning (CoFact)	Number of procurement orders Value of procured goods and/or services
	Procurement execution (CoFaith)	Number of new external contractors Maturity and fidelity of external contractors
	Procurement execution (CoInteraction)	Number of different organizations involved

5 A Case Study

The project used for the case study is a typical ICT project since it combines custom software development, procurement of Commercial Off-The-Shelf (COTS) software and hardware, and installation and operation of an electronic service. The system under evaluation is an Electronic Proposals Submission System that will facilitate the process of electronic proposal submission for the purposes of a Public Research Agency. More specifically, the project includes the following tasks: the development of a browser-independent web-based Proposals Submission System enabling research project proposals to be constructed and submitted electronically and the development of a system for preparing a proposal identical to those produced on the system via a stand-alone application.

In order to evaluate the complexity of the above-described project we will follow a qualitative approach that takes into account the complexity sources identified in Table 1. We will evaluate each subject area using a likert scale from 1 to 5, where 1 represents low contribution to project complexity and 5 represents significant contribution to complexity. Further, we make the assumption that all subject areas contribute equally to the complexity of the project, in our case 11.11%. However, this assumption can be easily changed with the application of a multi-criteria decision-making (MCDM) method, such as Analytical Hierarchical Process (AHP). The AHP method was introduced by Saaty [23] and its primary objective is to classify a number of alternatives (e.g., a set of candidate software metrics) by considering a given set of qualitative and/or quantitative criteria, according to pairwise comparisons/judgments provided by the decision makers. AHP results in a hierarchical leveling of the measurement criteria, where the upper hierarchy level is project management subject areas and the next level defines a set of complexity metrics, which can be further subdivided into subcriteria at lower hierarchy levels.

In our example, as it was mentioned previously, we have a simple scoring model with equal weights per subject area.

A key prerequisite for the calculation of the complexity is the definition of the scale that will be used for each metric. For example, the metric “*Number of change requests*” should be evaluated according to Table 2.

Having calculated a score for each contributing complexity factor (Table 3), we should proceed to the next step which is the calculation of the total project complexity. This calculation is presented in Table 4. Initially, we adjust the total score of the subject area according to the percentage that this subject area is participated in the total score. Subsequently, we take the sum of all partial scores.

In Table 4, in column A, % is given, since the organization assumes that all PM subject areas contribute equally in the project complexity. Column B is calculated by adding the scores of each complexity metric per subject area. Column C gives that total number of metrics used per subject area. Column D adjusts the score of column C, using 100 scale. Finally, column E calculates the total score using the contributing % of column A.

The total complexity score for this specific project is 72.93, a number that does not convey a lot of information unless we see it comparatively with other projects of the same organization.

Table 2 Example of metric’s scale

Complexity metric	Very low (1)	Low (2)	Moderate (3)	High (4)	Very high (5)
Number of change requests	1% requirements changes	3% requirement changes	5% requirements changes	10% requirements changes	>10%

Table 3 Calculation of complexity in three subject areas (sample)

Subject area	Indicative metrics	Justification	Score
Project Integration Management	Number of project management processes employed	Full life cycle is used. Further, system operation is required	4
	Level of maturity (e.g., CMM level)	System is critical for client organization. Therefore, a contractor with high process maturity should be selected	2
	Number of change requests	Requirements are stable and well defined	2
	Number of stated objectives	Limited number of stated objectives	1
	Financial resources committed	Financial resources committed 100%	1
	Number of phases	Traditional waterfall life cycle used	1
	Level of rigidity/agility		
	Number of similar software projects within organization	Moderate innovation in custom development	3
	Number of similar software projects in the literature	Similar implementations exist in the literature	2
	Legislation and regulations	Privacy and confidentiality legislation apply	4
Project Scope Management	Market and competition	Strong demand from customers	4
	Metrics such as FPA and UCP	Moderate number of FP (in comparison with other projects)	3
	Number of deliverables	Moderate number of deliverables	3
	WBS levels	Three WBS levels	3
	Number of components and modules	Small number of modules	2
	Number of third-party components used	More than 80% is based on COTS software	5
Project Time Management	Number of different technologies used within project	Moderate number of different technologies	3
	Project duration	Short project development duration	5
	Number of activities	Moderate number of activities	3
	Number of organizations (units) involved	Large number of units involved	4
	Number of different resources needed	Moderate	3
	Number of dependencies between activities	Low	2
	Number of activity constraints	Moderate	3

Table 4 Calculation of total project complexity

Subject area	Contributing (%) (A)	Unadjusted score (B)	Number of complexity factors (C)	Adjusted score (D)	Total score (E)
Project Integration Management	11.11	24	10	48	5.33
Project Scope Management	11.11	19	6	63	7.04
Project Time Management	11.11	20	6	67	7.41
Project Cost Management	11.11	7	3	47	5.19
Project Quality Management	11.11	7	2	70	7.78
Project Human Resource Management	11.11	30	7	86	9.52
Project Communication Management	11.11	20	5	80	8.89
Project Risk Management	11.11	15	3	100	11.11
Project Procurement Management	11.11	24	5	96	10.67

6 Conclusions

The need to measure complexity is well understood and sufficiently justified. Obviously, software project complexity is an area that needs to be studied further, and in detail.

In this chapter, we have presented a first set of ideas and a way to systematically measure complexity using the widely and well-known PMBOK framework. Of course, a lot of work remains to be done.

First, all presented elements have to be further analyzed in order to produce a model that is able to calculate project complexity robustly by combining factual, dynamic, and interaction elements, if possible, to produce a thermometer of complexity as it was proposed by Geraldi [7].

Second, we need to know how we can practically measure the evolution of project complexity over the project duration and what interventions are necessary for managing and controlling the complexity.

References

1. The Standish Group (2001) *Extreme Chaos*, The Standish Group [Online]. Available http://www.standishgroup.com/sample_research [Accessed: Dec. 19, 2008].
2. Charette, R. N. (2005) Software Hall of Shame, *IEEE Spectrum* [Online]. Available <http://www.spectrum.ieee.org/sep05/1685> [Accessed: Dec. 19, 2008].

3. Laird, L., and Brennan, M. (2006) *Software Measurement and Estimation. A Practical Approach*. John Wiley & Sons, Inc., Hoboken, New Jersey.
4. Florac, W. A., Park, R. E., and Carleton, D. (1997) *Practical Software Measurement: Measuring for Process Management and Improvement*. Software Engineering Institute (SEI), Pittsburgh, CMU/SEI-97-HB-003.
5. Simon, H. A. (1962) The architecture of complexity. *Proceedings of the American Philosophical Society*, Vol.106, No.6 (Dec.12, 1962), pp. 467–482.
6. Project Management Institute (2008) *A Guide to the Project Management Body of Knowledge*, 4th ed. Project Management Institute, ANSI/PMI Standard 99-001-2008.
7. Geraldi, J. (2008) Patterns of complexity: The thermometer of complexity. *Project Perspectives*, IPMA 29: 4–9.
8. Vidal, L. A., and Marle, F. (2008) Understanding project complexity: Implications on project management. *Kybernetes* 37(8): 1094–1110.
9. Cambridge Advanced Learner's Dictionary [Online]. Available <http://dictionary.cambridge.org>.
10. Park, R. (1992) Software size measurement: A framework for counting source statements. Carnegie Mellon University, CMU/SEI-92-TR-020 [Online]. Available <http://www.sei.cmu.edu/pub/documents/92.reports/pdf/tr20.92.pdf> [Accessed: Dec. 19, 2008].
11. Garmus, D., and Herron, D. (2001) *Function Point Analysis: Measurement Practices for Successful Software Projects*. Addison-Wesley. http://www.amazon.com/Function-Point-Analysis-Measurement-Addison-Wesley/dp/0201699443#reader_0201699443
12. Karner, G. (1993) *Metrics for Objectory*. Diploma thesis, University of Linköping, Sweden. No. LiTH-IDA-Ex-9344:21.
13. Boehm, B., Abts, C., Brown, A. W., Chulani, S., Clark, B. K., Horowitz, E., Madachy, R., Reifer, D. J., and Steece, B. (2000) *Software Cost Estimation with COCOMOII*. Prentice Hall, Englewood Cliffs.
14. Griffin, A. (1997) The effect of project and process characteristics on product development cycle time. *Journal of Marketing Research* 34: 24–35.
15. Baccarini, D. (1996) The concept of project complexity – A review. *International Journal of Project Management* 14(4): 201–204.
16. Camci, A., and Kotnour, T. (2006) Technology complexity in projects: Does classical project management work? *PICMET 2006 Proceedings*, Turkey, pp. 2181–2186.
17. Dooley, K. (2002) Organizational complexity. In: *International Encyclopedia of Business and Management*, M. Warner (ed.), Thompson Learning, London, pp. 5013–5022.
18. Geraldi, J., and Adlbrecht, G. (2007) On faith, fact, and interaction in projects. *Project Management Journal* 38(1): 32–43.
19. Geraldi, J. (2008) The balance between order and chaos in multi-project firms: A conceptual model. *International Journal of Project Management* 26: 348–356.
20. Williams, T. (2002) *Modeling Complex Projects*. Wiley, Chichester.
21. Whitt, S. J., and Maylor, H. (2007) And then came Complex Project Management (revised), *The Proceedings of 21st IPMA World Congress on Project Management*.
22. Richardson, K., Tait, A., Roos, J., and Lissack, M. R. (2005) The coherent management of complex project and the potential role of group decision support systems. In: *Managing Organizational Complexity: Philosophy, Theory and Application*, K. Richardson (Ed.), Information Age Publishing, Charlotte, NC, pp. 433–472.
23. Saaty, T. L. (1980) *The Analytic Hierarchy Process*. McGraw Hill, New York.

Application of Project Portfolio Management

Malgorzata Pankowska

Abstract The main goal of the chapter is the presentation of the application project portfolio management approach to support development of e-Municipality and public administration information systems. The models of how people publish and utilize information on the web have been transformed continually. Instead of simply viewing on static web pages, users publish their own content through blogs and photo- and video-sharing slides. Analysed in this chapter, ICT (Information Communication Technology) projects for municipalities cover the mixture of the static web pages, e-Government information systems, and Wikis. So, for the management of the ICT projects' mixtures the portfolio project management approach is proposed.

Keywords Project portfolio management · e-Municipality · Enterprise 2.0 · Web 2.0

1 Enterprise Architecture Development for Municipal Office

For further considerations of project portfolio management and exemplification, some introductive assumptions are necessary. According to ISO, enterprise is a group of organizations sharing a set of goals and objectives to offer products and services. Taking into account that definition, for now, the term of enterprise can be interpreted as an overall idea to identify a company, organization, business, or governmental institution [5]. Therefore a municipal office can be considered as an enterprise. The enterprise architecture is the process of translating business vision and strategy into effective enterprise change by creating, communicating, and improving the key principles and models that describe the enterprise's future state and enable its evolution. Enterprise architecture focuses on shaping and governing

M. Pankowska (✉)

Information Systems Department, University of Economics, Katowice, Poland
e-mail: malgorzata.pankowska@ae.katowice.pl

the design of the future enterprise using principles to stipulate future direction and models to underpin and visualize future states [9]. Considering enterprise architecture demands analysis and design of information systems. A software architecture defines a structure that organizes the software elements and the resources of a software system. Software architectures do not detail the internal structure of components, but they detail their externally visible behaviour and their relationships to other subsystems of the architecture. Municipalities as well as any enterprises are facing the challenge of integrating complex software systems in a heterogeneous information technology landscape that has grown in an evolutionary way for years. Most of the application systems have been developed independently of each other, and each application stores its data locally, either in a database system or some other data store. In enterprise computing, changes are abundant and system architecture should support changes in an efficient and effective manner. The enterprise application integration architecture resulting from point-to-point integration does not respond well to changes. The reason is due to the hard writing of the interfaces. Any change in the application landscape requires adaptation of the respective interfaces. This adaptation is typically realized by reprogramming interfaces, which requires considerable resources. A specific realization platform of enterprise application integration is message-oriented middleware, where applications communicate by sending and receiving messages [12]. The other solution is the implementation of Enterprise Resource Planning (ERP) system. An ERP system is an organization-wide software suite with integrated functionality covering all operational management areas of all organizational units. A similar definition was formulated in [4]. This enables an organization to, for example, link information on financial resources directly to information on other management aspects within one system. The integration of various functional areas within one system can provide many benefits to an organization. Some examples are improved connectivity with both other internal departments and external organizations due to standardization or improved opportunities to associate financial data with policy information. The ERP system for public sector has been elaborated by SAP software company [<http://www.sap.com>]. For central/federal, provincial/regional/state or local government, SAP provides innovative solutions that improve services while lowering costs. SAP application portfolio enables public organization to integrate processes across government departments, government levels with support for human resources management, government procurement, public sector accounting, social services and social security, government programmes, tax and revenue management, public security, and organization management and support. Originally ERP systems were developed for material production firms, however software houses are constantly looking for new clients, therefore they customize ERP system to implement them at governmental agencies, i.e. municipal offices. The Comarch software house (<http://www.comarch.pl>) is developing ERP systems for implementation in different sectors in Poland and abroad. The Comarch Egeria integrated management system is a modern, Polish class II ERP system supporting enterprise management. The system offers well-balanced functionality because it covers all the significant fields of an enterprise's operations. It is a universal tool that guarantees the stable

growth of every company and is also flexible enough to satisfy their diverse requirements. Each element of Comarch Egeria can operate as a stand-alone product and be deployed to expand the functionality of applications already in use.

Enterprise integrated information systems like SAP application portfolio are used to facilitate the seamless integration and data exchange between various departments within an organization. In order to achieve this, rigidly defined control mechanisms must be in place in the system, which safeguard the company's data and protect the system from unauthorized and unintended uses. This ideal for total control, however, is only achieved to a certain extent. Particularly, for ERP system control is centrally provided; however, within an enterprise, there are a lot of different applications, control, and protection which are highly decentralized.

The enterprise architecture identifies the main components of a municipal office and how components in the organization's nervous system function together to achieve defined public administration objectives. These components include personnel, administrative processes and procedures, technology, financial information, and other resources. If decisions about components and their interrelationships are uncoordinated, efforts and resources will be duplicated, performance and management problems will arise and will result in the loss of agility of the public administration organization. Enterprise architecture done correctly will ensure the choice of appropriate components and will specify how all components will operate together in a fluid manner that increases organizational efficiencies. Some issues should be taken into consideration when building system architecture to support the enterprise:

- The business processes that the applications, machines, and networks are supposed to support. This is the primary concern of the systems architecture for an enterprise. Hence, an architect needs to map all the municipal office's business processes to the applications and infrastructure that is supposed to support it.
- Any part of the overall architecture that is being compromised. Areas that often suffer are the data and security architecture and data quality. The data architecture suffers because users do not have the knowledge and expertise to understand the effects of their changes to procedure. Security architecture suffers because security is often neglected by employees. It should be noticed that at municipal offices employees have access to personal data of the citizens and that data ought to be protected as confidential.

2 Enterprise 2.0 as a New Opportunity for Communication Improvement

Although the Internet used to be a one-way communication channel, the Web 2.0 technologies now leverage the contributions of multiple participants. Web 2.0 is content-sharing sites, discussion and collaboration areas, and application design

patterns. They also represent a significant opportunity for organizations to build new social and web-based collaboration, productivity, and business systems and to improve cost and revenue returns [10]. Web 2.0 sites have become destinations for communities of users to create and share rich and complex data, such as music, images, and video and to discuss and rate the content. Web 2.0 represents a fundamental change in the way people interact with content, applications, and other users.

The concepts behind Web 2.0 strike a new balance between the control and administrative ease of centralized systems and the flexibility and user empowerment of distributed systems. One of the most visible features of Web 2.0 is the concept that the system is being constantly updated and enhanced, often in real time in response to user requests. The second important characteristic of Web 2.0 is the focus on data and content and in particular the ability for people to create and interact with rich content rather than just consume it. If the original Internet provided read access to data, then Web 2.0 is all about providing read and write access to data from any source. The original Internet was all about text, Web 2.0 started with music and images and moved into voice and video. Now TV and movies are the content areas that are being investigated as part of Web 2.0. While people and organizations have been searching, uploading, and downloading all this explicit data and content on the web, they have at the same time been creating a huge amount of implicit data about where they are going and what they are doing. The implicit data of Web 2.0 can be used to predict future behaviour or provide new attention-based features [13]. The third key element of Web 2.0 systems is the concept of social networks, community collaborative networks, and discussions. The unique element that Web 2.0 brings is that of social networks and community which are typically enabled by blogs, discussion groups, and Wikis. In Web 2.0 the people on the Internet create a social capital, where the interactions between people create information and systems that get better the more they are used and the more people who use them. In Web 2.0 people are grouped around project, event, content, idea, interest, person, and resource (material, spatial, financial).

While many of the best-known exponents of Web 2.0 such as Amazon, Facebook, and Google use proprietary software to manipulate and analyse the data, these services are also increasingly exposing their data using public Application Programming Interfaces (APIs), allowing third parties to combine multiple existing data sources to generate new and novel services, called mashups. Likewise, there are open standards that are commonly used to distribute blog content and podcasts, and programming models and techniques used to provide dynamic content and a rich visual experience to users [11].

The Association for Information and Image Management (AIM) defines Enterprise 2.0 as a system of web-based technologies that provide rapid and agile collaboration, information sharing, emergence, and integration capabilities in the extended enterprise [1]. Enterprise social software, known as Enterprise 2.0, is a term describing social software used in enterprise business context. It includes social and networked modifications to company intranets and other classic software platforms used by large companies, i.e. Oracle to organize their communications.

Table 1 Enterprise Information Systems vs. Enterprise 2.0 applications

Enterprise Information Systems	Enterprise 2.0 applications
Data oriented	Content oriented, i.e. data as well as music, images, figures, and photos
Database management systems	Content management systems
Internal application – users, i.e. enterprise employees	External application – Internet users
Developed according to well-established methodology	Agile software development
Commercially available software development tools	Open source software development tools
Traditional project management methods	Agile project management methods
Management information systems	Agile information systems
Transaction oriented systems, profit generating	Satisfaction-oriented systems, relations (social capital) creation
Business processes realization	Network extension
Material resources processing	Knowledge dissemination
Top-down governance of information	Bottom-up governance of information
Information usage for registration, transacting, searching, browsing, aggregating, modifications, and deleting	Information usage for publishing, subscribing, image creation, self-supporting, promoting, advertising, and evaluating
Information retrieval is for transactions	Information retrieval is for relationships creation
Information is aggregating for commercial purposes and high-level decisions making	Information is micro-aggregated, small portion of information for low-level decision making
Marketing and promotion is realized basing on ex-post information and information is pushed out, disseminated through traditional marketing channels	Marketing and promotion is based on conversations, personal online dialogues and web castings, etc.
Information control is ensured by the enterprise owner in their information security policy, well-defined information access profiles	Information is controlled by the content authors who are eventually responsible for what was disseminated on the web
Information is well structured in databases, registers, files, data warehouses, and data marts, included in forms and documents, web pages	Information is included in tagged objects
Software applications are commercially available, proprietary, closed	Software applications developed by the volunteers as open source, generally accessible standards based

In contrast to traditional enterprise software, which imposes structure prior to use, enterprise social software tends to encourage use prior to providing structure (Table 1).

Enterprise 2.0 describes the introduction and implementation of Web 2.0 technologies within an enterprise, including rich Internet applications, providing software as a service, and using the web as a general platform. Social software for

an enterprise allows users to search for other users or content, connect groups of similar users or content together, include blogs and Wikis, and allows for mashups for visualization. Enterprise 2.0 makes informal processes less chaotic and more transparent. It makes easier to capture informal knowledge and reuse it in other situations.

3 Exemplification of e-Municipality and Public Administration Information Systems Development

Emerging trends in Europe suggest that current thinking on e-Government is focusing on great quality and efficiency in public services. According to the view, e-Government needs to be more knowledge based, user centric, distributed, and networked. New public services will be required by the European Union (EU) as well as innovative ways of delivering the existing ones. Second, technological advances in the miniaturization and portability of ICTs suggest that in the future, e-Government will form part of an ambient intelligence environment, where technology will surround people and serve them on their roles as citizens, customers, and professionals. e-Government should have a strategic focus which includes the achievement of the Lisbon goals, the reduction of barriers to the internal market for services and mobility across Europe, the effective implementation of national policies, and regional or local development. Providing user-centred services and cutting unnecessary administrative burden require that information is shared across departments and different levels of government. The EU vision of e-Government places it at the core of public management modernization and reform, where technology is used as a strategic tool to modernize structures, processes, the regulatory framework, human resources, and the culture of public administrations to provide better government and ultimately increased public value [2].

In Poland, according to the European Commission, the Internet usage by individuals (15+) is 49% (2008), the following acts adapted by the Parliament (Sejm) were extremely important for e-Government and e-Democracy development:

- The Act on Computerization of the Operations of Certain Entities Performing Public Tasks of 17th February 2005. The act sets up horizontal/infrastructure programmes for all the sectors of public administration and establishes a common interoperability framework for ICT systems in the Polish public sector.
- New law on Public Procurement of 29th January 2004, enabling the development of e-Procurement systems for Polish public administrations and allowing the use of electronic auctions for contracts up to €60000.
- The strategy on the Development of the Information Society in Poland for the years 2010–2013 prepared by the Ministry of Scientific Research and Information Technology.
- The launch of the Public Information Bulletin (PIB) (official electronic journal on public information) in accordance with the Act on Access to Public Information of 6th September 2001. The Ministry of Internal Affairs and Administration is responsible for the PIB.

The right of access to public information constitutes a major component of the democratic standard of open government or openness of public authorities and entities responsible to them. The openness is founded on the transparency of organizations and their operations. Cyberspace represents a place where people can communicate to exchange views and to develop socio-economic and political initiatives. Through new venues, people can engage in many sorts of economic and political activities, such as joining interest groups, voting in elections, or participating in forums to solve joint problems that appeared in their town or province. For this chapter, the content analysis was conducted on a sample of municipality web sites to provide empirical validation for the deliberativeness for the new socio-economic spaces. Content of the web pages and forums' postings comprise a defined context or horizon from which citizen municipality discussion and collaboration can be evaluated. It is not necessary to know who the participants are to present the conclusions on e-Administration and virtual community development. Content analysis covers web sites for small communities (i.e. Strykow, 3000 inhabitants, <http://www.strykow.pl>) as well as for big cities (i.e. Poznan, 570,000 inhabitants, <http://www.poznan.pl/>, Cracow, 769,000 citizens, <http://www.krakow.pl>). Full list of the addresses of the analysed web sites is available upon request.

Resolution of the Act on Access to Public Information of 6th June 2001 caused rapid development of electronic publications named Public Information Bulletin (Biuletyn Informacji Publicznej, BIP). It is estimated that 98% communities in Poland develop and implement this electronic publication, although only 90% of them have constructed an official web site for the community. Content of Public Information Bulletin (PIB) is similar for municipalities, usually it contains information important for citizens on community authorities, organization of community offices, e-forms of documents, administrative procedures mandatory for citizens and for local businesses, declaration of private properties owned by community authorities, community legal regulations and rules, information on procurements for community, invitations for tenders and auctions, community budgets, and land planning. Generally, community web sites and PIBs ensure achievement of e-Administration goals as it is assumed in Information Society Development Strategy for EU countries. They enable transfer of top-down administrative information and access to governmental sources of public information, as well as to portals for law interpretation and public administration knowledge dissemination. The citizens have the opportunities to learn about legal acts mandatory for them, to recognize office procedures. They can download forms of documents. They have the possibility to utilize multi-channel communication with Citizen Service Office, where an official can use stationary telephone, mobile, and e-mails. However, citizens still mostly prefer F2F (face-to-face) contacts. The PIBs usually ensure investors' and business units' access to databases of tenders considering jobs for public institutions.

For public administration clients, the Internet is providing information of two kinds. One is mandatory public information such as pertaining to laws, departmental operation, formal procedure requirements. The other consists of personal experiences that are reported by individuals voluntarily. Public information is provided in

the official Public Information Bulletin, but virtual communities can be developed for gathering and exchanging personal opinions and impressions. The virtual community can be interpreted as a communication medium influencing the personal networks of inhabitants of a neighbourhood within a municipality. Another view is the virtual community as a tool to improve local democracy and participation; in fact it is the basic idea behind the digital city in Amsterdam [8]. Virtual community develops as an experiment with new forms of solving problems and coordinating social life. As a free space to experience and to exchange views, virtual community requires ICT tools, i.e. e-mailing, www conferencing, announcement email distribution list, citizens' open discussion forum, and newsgroups.

The study of the content of the forums allows for the conclusion that they are not sufficiently well utilized as a medium to involve citizens and officials to act for the community. Within the forum, people need a leader to conduct the discussion, as for example mayor of the town, or a problem which ought to be solved successfully for all the stakeholders involved in the discussion. Shared interests of the people involved in the same virtual community integrate them more than living in the same building. People have strong commitments to their online groups when they perceive them to be long-lasting. There is a danger that virtual communities may develop homogeneous interests. It must be noticed that people do not see that the Internet is especially suited to maintaining intermediate-strength ties between people who cannot visit each other frequently. Online relationships are based more on shared interests and less on shared social characteristics. Living in the same non-attractive village is not a sufficient argument for discussion in the forum. The limited evidence available suggests that the ties people develop and maintain in cyberspace are much like most of their real life community. In big cities (i.e. Poznan) people noticed the advantage of communication in forum. They got used to communicating online, but in small towns (i.e. Krosno <http://www.krosno.pl>) people know how to use it, but they do not see opportunities for application of forum, chat, blogs. Instead of forum, people ask for photo gallery (e.g. Nysa, <http://um.nysa.pl>). Films and photo pictures are more impressive and persuasive than text to integrate people around problems or latest events in community (e.g. Wroclaw, <http://www.wroclaw.pl>, Rybnik <http://www.rybnik.pl>). Photo and film galleries seem to be a natural way to integrate citizens in virtual communities. It is worth to notice that the web derives analytically not technically from two important stands of media: broadcast media like radio and television and individual communications media like the telephone. As a medium, it holds the potential to incorporate previous focus of mass media – television, audio, radio, text, and photography – and combine them with the interactivity of the telephone. In comparison with the previous media, the Internet has these advantages and it enables citizens to present themselves and view themselves. The opportunity to see themselves on the Internet around the family, neighbours, friends, and others more or less known create an irrefutable impression that the entire world has the possibility to view the film or photos. Virtual community focused on photo gallery and the community news develops a potential to stimulate visitors' imagination and interests by engaging in creative communication while simultaneously presenting them with a fantastic array of visual and auditory sensations. Digital technologies cannot be regarded as the panacea to many of the

problems which underline the apparent civic disengagement. The use of information and communication technologies and strategies by democratic actors (government, elected officials, the media, political organizations, citizens/voters) within political and governance processes on national and international stage requires long-term education. People may know themselves, but they actively want to maintain community ties. Intensive relations mean for them the belonging to the community. The belonging depends on four systems:

- civic integration that means being an empowered citizen in a democratic system;
- economic integration that means having a job and a valued economic function;
- social integration that means having access to the state support without stigma;
- interpersonal integration that means having family, friends, neighbours and social networks.

Although Web 2.0 social software supports creation of virtual communities and informal communication processes development, governmental agencies are involved in further realization of Lisbon strategy and Information Society Development Strategy. Therefore, EU funds are spent for the implementation of public administration information systems, i.e. SEKAP system. SEKAP (<http://www.sekap.pl>) is the electronic communication system for public administration in Silesia, Poland. That EU-funded project of the local and regional authorities was realized in 2005–2007 with the objective to deliver easy access to information and public e-Services, which are a good way of implementing the information governance model. The information strategy within the project was quite well specified and concerns enabling access to public administration electronic services for citizens and transfer of electronic documents among municipal and regional public administration agencies in 54 towns in Silesia, Poland. The ICT architecture for SEKAP system is sufficiently well developed, including the customized software and hardware. The SEKAP software comprises docflow and workflow system integrated with the Public Information Bulletin, e-forms platform for citizens, public services, automatic digital signatures verification system, security system, and e-payments system. The SEKAP hardware covers data centre equipment, individual infrastructure, and digital signature equipment.

4 Project Portfolio Management for ICT Implementation at Municipal Offices

In order to meet its goals, every organization launches multiple projects during a fiscal year. Some projects may have dedicated resources, resources that work on only one project at a time. However, more commonly some resources are used across many projects. Moreover, people are often assigned multiple projects at the same time. The likelihood of finishing projects on time, on budget, and within scope is low. Multiple project development work requires coordination and communication and sometimes this may lead to changes in requirements, resource availability, and in the detailed schedule. These changes can place the best-organized project team

in dire jeopardy, leaving the team to work in high-stress situations that raise the delivery risk even higher. Although for many local communities management of group of projects is very stressful, they understand the need to develop the project portfolio management methods. Project portfolio management must be strategically oriented and concentrates on people, who are project stakeholders; on processes and procedures for project goals achievements; and on creation and usage of project techniques and tools.

Taking into account the analysis of Polish municipal communities web sites content, the following project can be included in the project portfolio at a municipal office:

- Enterprise Resource Planning (ERP). The ERP project is initiated to identify an enterprise-wide solution to replace and enhance functionality provided by the aging multiple non-integrated systems currently performing the city's core business activities. Phase I of the ERP implementation included financial and accounting systems, human resources, payroll, purchasing, budget, asset management, project and grant accounting, and tax collection.
- Geographic Information System (GIS). The demands of constituents, both internal and external, are exceeding the capabilities of the city's current department-oriented, geographic data files and storage structure.
- Computer-Aided Dispatch System (CAD). This project studied the different, disparate computer-aided dispatching and records management systems in use in the city.
- Municipality Project Portfolio Management (PPM) system. The planned deployment may include project management process re-engineering and training and the use of the Project Knowledge System (PKS).
- Public Information Bulletin as static web pages covering the regulations and the official publications of municipal offices.
- Links and place of access to the regional public administration information systems, i.e. SEKAP.
- Multimedia repositories to support Wikis development co-funded by municipality and citizens.

Portfolio management ensures that the collection of projects chosen and completed meets the goal of the organization, just as a stock portfolio manager looks for ways to improve the return on investment, so does a project portfolio manager. However, the character of ICT developments implies that ICT can have an effect in almost every aspect of an enterprise. ICT can enable significant competitive benefits, i.e. strategic match, competitive advantage, and improving the information management. Project portfolio management has some important responsibilities:

- Determining a viable project mix, that is capable of meeting the organization goals;
- balancing the portfolio, to ensure a mix of projects that balances short term vs. long-term investment, their risk and rewards, research and development;

- monitoring the planning and execution of the chosen projects;
- analyzing portfolio management and ways to improve it;
- evaluating the new opportunities for further projects development, taking into account the organizational, project execution capabilities [7].

Nowadays, project portfolio management is supported by Val IT, ISACA framework focusing on the rationality of investment decisions and the realization of benefits. Val IT provides guidelines, processes, and supporting practices to assist the executives in understanding and carrying on a group of projects [3]. There are strong differences among financial portfolio and ICT project portfolio [6]. The latter constitutes the software components that are not substitutive, but as complementary and interdependent, they are used to deliver competitive advantage to business by providing various services and capabilities. The ICT assets resulting from ICT project portfolio are gathered together to maximize the business value, i.e. reduction of information processing costs, increase of customer satisfaction, improvement of business processes throughput.

5 Conclusions

Generally, project portfolio management is a continuous process, although a periodic evaluation of vendors, technologies, and risks is a critical factor. Project portfolio management can ensure sustainable approach to municipal office application inventory with consistent business metrics, stakeholders' participation for data gathering and ongoing assessment, automation of public administration processes, participation of citizens in local community information management, and the ability to manage ICT projects in the context of all project portfolios on municipal level. As it was shown in this chapter, the ICT project portfolio approach can be applied at the municipalities' level to increase the value for citizens. The ICT assets belong to the portfolios because of their functionalities. However, their exchangeabilities are limited because the procurement of the software, hardware, and facilities components is a relatively slow process, information systems require time for their implementation, and they demand verification and stabilization before the beginning of the delivery of the expected effects. As it was presented, some portfolio components are gathered for free as it is in the case of open source software acquisition. Others are created by end users, i.e. Wikis. Anyway they all together provide the business value. At the municipal level, the centralized management of a group of ICT projects within portfolio is particularly important from the point of view of project controlling and project information distribution and accessibility.

References

1. Bonham S. S. (2005) *IT Project Portfolio Management*. Artech House, Boston, London.
2. Centeno C., van Bavel R., and Burgelman J.-C. (2004) *eGovernment in the EU in 2010: Key policy and research challenges* (available: <http://www.jrc.es/home/publications/publications.html>).

3. Enterprise Value: Governance of IT investments, The Val IT Framework (2006) IT Governance Institute, Rolling Meadows (available: <http://www.isaca.org>).
4. Esteves J., and Pastor J. (2001) Enterprise resource planning systems research: An annotated bibliography. *Communication of AIS* 7(8): 1–52.
5. Hoogervorst J. P. (2009) *Enterprise Governance and Enterprise Engineering*. Springer, Diemen.
6. Kasargod D., and Bondugula K. Ch. (2005) *Application Portfolio Management, A Portfolio Approach to Managing IT Applications Can Help Banks Improve Their Business Performance*. Infosys Technologies Limited (available: <http://www.infosys.com/offering/industries/banking-capital-market/APM-strategy-for-banks.pdf>)
7. Kendall G. I. and Rollins S.C. (2003) *Advanced Project Portfolio Management and the PMO: Multiplying ROI at Warp Speed*. Ross Publishing, Boca Raton, FL.
8. Melis I., van den Besselaar P., and Beckers D. (2000) Digital cities: Organization, content and use. In: Toru Ishida and K. Isbister (Eds) *Digital Cities: Experiences, Technologies and Future Perspectives*, Lecture Notes in Computer Science 1765. Springer-Verlag, Berlin, pp. 18–32.
9. Op't Land M., Proper E., Waage M., Cloo J., and Steghuis C. (2009) *Enterprise Architecture, Creating Value by Informed Governance*. Springer, Berlin Heidelberg
10. Platt M. (2009) Web 2.0 in the Enterprise. *The Architecture Journal* (available: <http://msdn.microsoft.com/en-us/library/bb735306.aspx>).
11. Taylor I. J., Harrison A. B. (2009) *From P2P and Grids to Services on the Web*. Springer, London.
12. Weske M. (2007) *Business Process Management, Concepts, Languages and Architectures*. Springer, Berlin Heidelberg.
13. White B. (2007) The implications of Web 2.0 on web information systems. In: Filipe J., Cordeiro J., and Pedrosa V. (Eds) *Web Information Systems and Technologies, International Conferences, WEBIST 2005 and WEBIST 2006 Revised*. Springer, Berlin, pp. 3–8.

Part III
Human-Computer Interaction and
Knowledge Management

Towards a Cost-Effective Evaluation Approach for Web Portal Interfaces

Andrina Granić, Ivica Mitrović, and Nikola Marangunić

Abstract A cost-effective approach for web portal usability evaluation is presented in this chapter. Due to specifics of portals as web sites, mainly referring to their structure and media specificities along with diversity of users, tasks and work-flows, distinct assessment approaches should be employed. Methodology brings together laboratory-based testing along with experts' inspection and produces valuable results for users and developers at a low cost. Compared to our first study, user assessment applied a faster and less expensive procedure, providing stability of measures with reduced sample size, while inspection employed fewer specialists with higher expertise and a simpler evaluation form. Directions of future work are identified.

Keywords User testing · Guideline inspection · Cost-effective evaluation approach · Web portals

1 Introduction

Current research on usability evaluation in general clearly searches for methods that produce beneficial results for users and developers at a low cost, perhaps with the economic of assessment as the most important factor, cf. [14, 19]. The aim of the overall research is the design of a cost-effective methodology for web portal interface evaluation. This chapter reports on just one part of this comprehensive research addressing design and results of the experimental study of news portals.

The main motivation for this research initiative came from reports stating that broad-reach and news portals are the most visited Croatian web sites. In order to evaluate how efficient and easy to use those portals are, we conducted an experiment employing a range of assessment methods, both empirical and analytic. The proposed evaluation guides in conducting focused assessment activities, obtaining

A. Granić (✉)
Faculty of Science, University of Split, Split, Croatia
e-mail: andrina.granic@pmfst.hr

useful data and producing helpful usability information in cost-effective way. Methodology brings together two basic evaluation approaches, inspection method and user assessments which embody an integration of four empirical methods into laboratory-based usability testing. Our experience indicated that the complementing experimental methods from the scenario-based testing proved to be consistent. Compared to our first study [10], assessment was made cost-effective by applying a faster and less expensive procedure which provides stability of measures with reduced sample size. Furthermore, inspection evaluation produced valuable results at a low cost with an employment of a fewer number of specialists with higher expertise and a simpler evaluation form. Nevertheless, much more research and analysis is needed in order to interpret an interesting finding in this study – a disagreement of obtained results from the testing and inspection methods.

The chapter is organized as follows: Section 2 introduces basic concepts and the aim of the overall research. Section 3 addresses the proposed cost-effective approach to web portal evaluation, offering an insight into related data collection and analysis. Section 4 brings discussion and concludes the chapter.

2 Aims and Basic Concepts

A web portal can be defined as a single point of access to information, resources and services covering a wide range of topics [29]. Through the blend of information, services and collaboration among users, a portal's primary objective is to create a working environment that the users can easily navigate through. Web portals are far more complex than "common" web pages, offering almost "self-contained" working environment with user-specific, customized views. For example, a university's portal could offer personalized content based on user roles (e.g. staff, student and administrator). The portal uses the information stored in the roles to offer the appropriate content and service choices. The services provided by portals could be classified into three categories – search, information and personal services [23], and authors argue that these three different functions affect portal use in different ways.

Market research findings related to the Croatian web context report that nowadays broad-reach and news web portals are the most visited Croatian web sites cf. [8]. Broad-reach portals offer a collection of services such as search engines, online shopping, e-mail, news, forums, maps, event guides, employment, travel and other kinds of information. On the other hand, news portals present information regarding latest news and events. Although they are often considered as specialized, concerning their structure and media specificities, news portals are very similar to the broad-reach ones. It has been argued that major Croatian broad-reach portals are reminiscent of the web sites of broadsheet newspapers or public service broadcasters [25]. A number of news portals today work as an engine to gather news from web sites of news agencies and newspapers organizing them in one single place. They came to be called "multi-source news portals" [5]. News portals inevitably replace paper-based/print editions which constantly decline, thus undertaking the role of

mainstream media and becoming the leading media for informing. For example, web site <http://www.newspaperdeathwatch.com> follows the decline of print newspapers (US dailies that have closed) and is looking for former print dailies that have adopted hybrid online-print or online-only models.

Information presented on each page of a web portal addresses a very large user group with highly diverse needs and interests, aspects which have to be reflected by the portal design. Thus an easy-to-use portal should be built with an understanding of the tasks users perform on a regular basis. It has to determine the steps or processes conducted by their users and then integrate utilities of services to improve a workflow [6]. In that context an effective web portal user interface assessment is essential because it identifies design problems to be corrected and also provides guidance for the next iteration of the development process. In general, usability, as quality of use [1], is context dependent and is shaped by the interaction between users, tasks and system purpose, e.g. [12]. A variety of usability evaluation methods have been developed over the past few decades and most used ones are grouped into two categories, e.g. [18, 21],: (i) usability test methods; user based which involve end users, hence include user testing, focus groups, interviews, questionnaires and surveys as well as (ii) usability inspection methods, without end users embracing heuristic evaluations and cognitive walkthroughs as frequently used ones. Recent research has had a tendency to bring together those two basic approaches, cf. [17]. Due to emphasized specifics of web portals as web sites, here primarily addressing their structure and media specificities along with diversity of user population, their tasks and workflows, particular usability evaluation approaches should be employed.

Current research on usability evaluation is mostly related with focused portals such as enterprise portals [2], travel portals [7], library web portals [3], tourist portals [17], healthcare portals [24] and similar. Moreover, research related to the evaluation of news portals has been fairly small, where the study aimed to identify areas of web usability in the news portal industry that may be culturally specific could be stressed [27]. Consequently, acknowledging the importance and necessity of web portal interface evaluation and at the same time overcoming the lack of related research, we have designed a cost-effective approach for web portal assessment. The focus of this chapter is on a single segment of the conducted research which addresses the experimental study related to the most visited online versions of the Croatian print-based newspapers.

3 Cost-Effective Approach to Portal Evaluation

In the context of our research, the first study included the assessment of broad-reach portals, both through a number of test methods and the usability inspection method, see for example [10]. Our experience indicated that the chosen research instruments, measures and methods for user evaluations were consistent. However, user testing could be improved by employing a faster and less expensive procedure involving fewer test users. Additionally, particular aspects of the inspection method could be

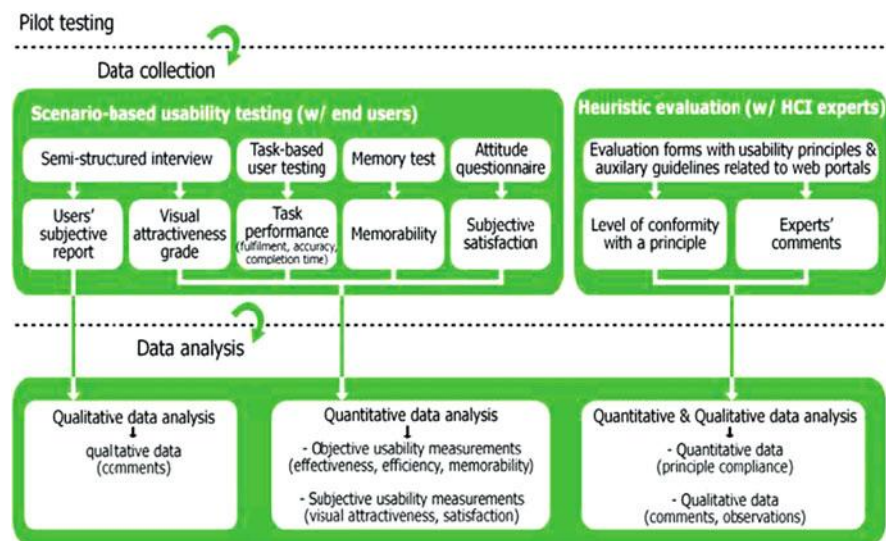


Fig. 1 Comprehensive cost-effective evaluation approach for web portal interfaces

upgraded as well; here referring to the issues of the experts' selection as well as the applied evaluation form which showed poor applicability in web portal context (did not provide useful information).

Therefore, it seemed valuable to perform a new usability assessment but now considering different web portals, valuating at the same time the improved usability evaluation approach. The discount evaluation approach for web portal interfaces was used in the assessment of the second type of most visited Croatian web sites – news portals, in some way specialized but still very similar to the broad-reach ones, as already clarified. The proposed assessment guides in conducting focused assessment activities, obtaining diverse and useful data, facilitating proper data interpretation and producing high-quality usability information in cost-effective way. Methodology brings together two basic evaluation approaches, inspection method and user assessments which embody an integration of four empirical methods into laboratory-based usability testing. It is consisted of two phases which are illustrated on Fig. 1 and described in the following subsections.

3.1 Data Collection

In the phase of data collection, different types of raw usability data are collected by means of four empirical methods, such as task performance data, memory data, rating results of attitude questionnaire along with users' subjective report from the interview. Furthermore, experts' reports along with their level of conformity with a guideline are gathered via guideline inspection.

3.1.1 Scenario-Guided User Evaluations

We conducted a controlled experiment which advocates scenario-guided user evaluations involving a number of usability testing methods used to collect both quantitative data and qualitative “remarks”. This methodology triangulation of *four empirical methods* involved 16 participants, 10 male and 6 female randomly chosen computer science students from the third and fourth graduate year, aged from 20 to 23 years. This was a representative group of participants because market research findings report that the majority of knowledgeable Croatian Internet users are students, aged from 15 to 24 years [9].

End-user testing was based on criteria expressed in terms of two types of measures: (i) objective performance measurement of effectiveness, efficiency and memorability as well as (ii) subjective users’ assessment, cf. [15]. The System Usability Scale (SUS), a simple, standard, 10-item attitude questionnaire with a 5-point Likert scale [4], was used for the subjective evaluation. As additional subjective feedback, answers to the semi-structured interview were collected. We included four news portals: Slobodnadalmacija portal (www.slobodnadalmacija.hr), Jutarnji portal (www.jutarnji.hr), Vecernji portal (www.vecernji.hr) and 24sata portal (www.24sata.hr).

In order to understand the effect of portal design in a sample work situation, we described a work scenario, a sequence of typical tasks and user actions. To test assigned tasks and time interval, clarity and unambiguity of measuring instruments for subjective assessment along with adequacy of hardware and software support, pilot testing was performed. We chose several typical tasks whose structure and location on the portals had not changed over time. The tasks, which covered different topics, were categorized in four categories: fact finding, information gathering, browsing and transactions [16]. For each portal selected, the tasks undertaken were the same and the probability of their completion was similar. The evaluation procedure was carried out individually with each test user, using a personal computer with Internet access in addition to software and hardware support for tracing and recording users’ actions and navigation. Within each evaluation session all the portals were assessed, with the order of their evaluation randomly selected. An evaluation procedure consisted of the following steps: task-based user testing, memory test, attitude questionnaire and semi-structured interview.

Task-based user testing involved a scenario-guided user assessment with tasks selected to show the basic portal functionality. It enabled us to determine user efficiency (time on task) and effectiveness (percent of task completed) while working. A user’s objective accomplishment measure, labelled as fulfilment, was calculated as the average time spent on all allocated tasks weighted with successfulness of task completion. For each user, the time limit for all assigned tasks was 15 min per portal. A *memory test* was performed subsequently to the task-based test and enabled the measurement of interface memorability by requiring a user to explain the effects of a single action or to write down the name for a particular operation. An *attitude questionnaire* enabled the assessment of the users’ subjective satisfaction with diverse types of interaction. We used the SUS questionnaire, as it is argued that this

yields the most reliable results across sample sizes [28]. Its questions address different aspects of user's reaction to the portal as a whole, providing an indication of the level of statement agreement on a five-point Likert scale. The feedback was augmented with the users' answers in a *semi-structured interview*. In this interview we asked the participants to rate and comment on the portal's visual attractiveness too.

3.1.2 Guideline Inspection

User evaluations were supplemented with less strict heuristic evaluation [22], i.e. *guideline inspection* conducted with a group of four "instant" specialists from the HCI field. With the intention of overcoming the problem of not having enough usability experts who could be involved in the evaluation, we had the inspection performed by "instant experts" [31]. They were mostly web design practitioners with years of experience in portal interface design who learnt the principles of good, user-centred design and provided expert assessment. An evaluation form consisting of a set of principles/guidelines augmented with auxiliary guidelines as additional explanations related to portals was prepared. Individual expert's marks and comments concerning the assessed portals were collected. The score for every portal was calculated as an average mark on a seven-point Likert scale. The same four portals were included in the study.

A document containing detailed instructions and an evaluation form was sent to chosen experts. Aiming to discover possible problems in the interface design, they had to mentally simulate the tasks to be performed on portals, mark and comment on the evaluation form, following the instructions and the provided guidelines along with the auxiliary ones. Thus, in order to supply all necessary information, the evaluation form had to be very detailed and self-explanatory. A set of seven guidelines, not so strictly based on Nielsen's [22] heuristics but more suitable for a portal context, was explained. Besides, as additional explanation to the guideline, a series of auxiliary guidelines concerning portal design were also provided, cf. [20, 30]. Experts had (i) to specify a level of their conformity with a principle and related set of auxiliary guidelines on a seven-point Likert scale and (ii) to provide a comment to justify the mark they assigned. They were encouraged to offer additional notes related to advantages and disadvantages of portals. Also observations and remarks concerning the overall evaluation procedure were welcomed.

3.2 Data Analysis

Results acquired through the usability test methods in addition to the main findings obtained in guideline inspection are addressed in what follows.

3.2.1 Analysis of Usability Problems from the User Evaluations

Descriptive statistics of the objective accomplishment measure fulfilment, including arithmetic means, standard deviations and significance levels of Kolmogorov – Smirnov coefficient for normality of distribution, are presented in Table 1. No

Table 1 Results of objective accomplishment measure fulfilment for the four selected portals (note that lower *M* score means better result)

Fulfilment	<i>M</i>	SD	K–S	df	<i>F</i>	<i>P</i>
Slobodnadalmacija portal	95.32	28.951	0.668	15	412.88	<0.01
Jutarnji portal	70.94	24.212	0.639			
Vecernji portal	56.88	13.883	0.245			
24sata portal	75.40	17.265	0.951			

statistical difference in the distribution of the results from the expected normal distribution was found ($K - S_{1,2,3,4} > 0.05$). In order to test the difference among portals, the analysis of variance (one-way ANOVA) as a parametric procedure was applied. Significant *F*-ratio ($F = 412.88$, $df = 15$, $p < 0.01$) indicates the existence of differences among the portals in the results related to the obtained objective measure. Post-hoc tests showed significant difference for all portal combinations except between Jutarnji portal and 24sata portal.

Descriptive statistics of the objective accomplishment measure memo, including arithmetic means, standard deviations and significance levels of Kolmogorov – Smirnov coefficient for normality of distribution, are presented in Table 2. No statistical difference in the distribution of the results from the expected normal distribution was found ($K - S_{1,2,3,4} > 0.05$). In order to test the difference among portals, one-way ANOVA as a parametric procedure was applied. Significant *F*-ratio ($F = 283.43$, $df = 15$, $p < 0.01$) indicates the existence of differences among the portals in the results related to the obtained objective measure. Post-hoc procedure revealed differences between Slobodnadalmacija portal and Jutarnji portal, in addition to Vecernji portal and Jutarnji portal.

Descriptive statistics of results acquired for subjective satisfaction, measuring the SUS for each participant on every web portal, are shown in Table 3. Again no statistical difference in the distribution of the results from the expected normal distribution was found ($K - S_{1,2,3,4} > 0.05$) and the difference among portals was tested using one-way analysis of variance. Significant *F*-ratio ($F = 597.65$, $df = 15$, $p < 0.01$) indicates the existence of differences among the portals in the results related to the obtained subjective measure (see Table 3). Those differences were found in post-hoc procedure only between Slobodnadalmacija portal and Vecernji portal, as well as Vecernji portal and 24sata portal.

Table 2 Results of objective accomplishment measure memo for the four selected portals (note that higher *M* score means better result)

Memo	<i>M</i>	SD	K–S	df	<i>F</i>	<i>p</i>
Slobodnadalmacija portal	4.03	2.061	0.944	15	283.43	<0.01
Jutarnji portal	2.78	1.291	0.660			
Vecernji portal	4.63	1.821	0.127			
24sata portal	3.50	1.438	0.554			

Table 3 Results of subjective satisfaction measure SUS for the four selected portals (note that higher M score means better result)

SUS	<i>M</i>	SD	K–S	df	<i>F</i>	<i>p</i>
Slobodnadalmacija portal	57.97	22.989	0.419	15	597.65	<0.01
Jutarnji portal	73.59	17.175	0.494			
Vecernji portal	80.78	12.066	1.000			
24sata portal	58.75	29.040	0.468			

Pearson’s correlation coefficients for the participants’ results in the achieved usability objective and subjective measures showed significant correlation between SUS and fulfilment on Slobodnadalmacija ($r = -0.61$, $p < 0.05$) and Jutarnji portal ($r = -0.57$, $p < 0.05$). Significant correlation was also found between SUS and visual attractiveness on 24sata ($r = 0.69$, $p < 0.01$) and Jutarnji portal ($r = 0.52$, $p < 0.05$). No significant correlation was revealed between any results on other portals.

3.2.2 Data Analysis from Guideline Inspection

Arithmetic means of marks from a seven-point Likert scale provided by four specialists according to seven usability guidelines show that the highest mark was given to 24sata portal (mean = 5.89), followed by Slobodnadalmacija portal (mean = 5.07), Jutarnji portal (mean = 4.86) and Vecernji portal (mean = 4.81).

The guidelines were “horizontally” examined through expert’s comments and observations, assigning low (L), medium (M) and high (H) values according to the quantity and the level of details of comments provided (see Table 4). “Vertical” analysis comprised an inspection of the specialist’s answers to the guideline compliance related to the assessed portals (see Table 5).

Table 4 Analysis of the adapted guidelines

Portals	Slobodnadalmacija portal		Jutarnji portal		Vecernji portal		24sata portal	
	Mark span	Info	Mark span	Info	Mark span	Info	Mark span	Info
1	5–6	H	4–6	M	2–7	M	6–7	M
2	5–7	H	3–6	H	5–7	M	4–7	H
3	5–6	H	4–6	M	5–6	H	3–7	H
4	5–7	M	5–7	M	5–7	L	4–7	L
5	3–5	H	4–5	M	3–5	M	6–7	H
6	2–7	M	4–5	H	3–5	L	3–7	M
7	3–4	M	2–5	L	3–6	M	5–6	L

1, Concept of portal is well adjusted to the user context; 2, while working with portal users have a feeling of control, safety and navigation freedom; 3, portal respects media standards and usual practice/usage/routine; 4, user gets information intuitively on the portal, i.e. user does not have to remember the information path but recognize it; 5, portal is adjusted for efficient use by novice users as well as by experts; 6, portals’ design is clear, understandable and transparent, i.e. most needed information are at the same time most visible; 7, portal offers help while working on it.

Table 5 Analysis of the experts’ feedback

Expert ID	Comments			Additional observations
	Number	Percentage	Quality	
1	25	89	M	None
2	28	100	M	3
3	28	100	H	3
4	28	100	H	5

Furthermore, we considered the outcomes achieved in the guideline evaluation in addition to the ones obtained throughout the usability testing. The ranking of the selected broad-reach web portals was compared. The result of usability inspection achieved did not conform to those obtained through applied usability testing. The highest ranked web portal in the end-user testing scored as the lowest one in the expert evaluation.

4 Discussion and Conclusion

The results of the scenario-guided user evaluation showed statistically significant differences among the portals according to the measures of user’s objective achievement, memo and fulfilment. This suggests that portals could be ranked by mean values. The results of the subjective satisfaction measure SUS also showed differences among portals and their ranking by mean values. Taken as a whole, significant correlations between objective and subjective measures on two portals were identified, indicating some equal factors that affect these different measures. It is important to point out that the measures of user’s objective accomplishment and her/his subjective satisfaction were not significantly correlated on other two portals. Achieved results are in accordance with the ones of the meta-analytic research report on correlations among usability measures calculated from the raw data of 73 studies [13]. Consequently, although relationship between objective and subjective measures has been shown, in order to get more precise insight of the interface “look and feel” both should be used (fulfilment and SUS measures as the essential ones).

The overall achieved results could be further related to the most frequent statements from the interviews. The participants felt especially pleased and comfortable working with the portals where their objective achievement was high. They considered them as sites with good quality of information structure, clarity and straightforward orientation. Correlation between SUS results and visual attractiveness on two portals indicates that a pleasant appearance influences the subjective perception of portal usability. The interview statements also support this finding. The participants usually emphasized portals’ visual attractiveness, assigning high subjective ratings. Such assumption is in line with related studies which also address

aesthetics aspects of design, cf. [11, 26]. Due to the identified statistical differences among portals in the fulfilment, memo and SUS, portals could be ranked according to the objective and subjective measures.

Statistical post-sample analysis conducted in our prior research showed stability of measures with 15–17 participants, so the original end-user sample of 30 participants has been cut in half. Therefore a more cost-effective and reliable approach has been obtained (i.e. the same portal is shown to be best for the three measures). Our experience suggests that the choice of the sample size in addition to the structure of engaged end users in conducted usability testing is also in accordance with the outcomes of related studies. Specifically, in the Hornbæk and Law's [13] meta-analysis of usability measures, the average number of participants involved per study was 32 ($SD = 29$, ranging from 6 to 181).

The very interesting finding of the conducted study is that the testing and inspection methods disagree. The achieved results of guideline inspection did not conform to those obtained through applied user evaluation. The highest ranked web portal by users scored as the lowest one in the specialist evaluation. These findings reflect concerns which have been already raised. On the one hand, it could be argued that the reason is in the guidelines adaptation to portal context along with the selection of HCI specialists who in fact were mostly web design practitioners. Alternatively, users understanding of “quality of use” concept could be different from the specialists' conception. Or possibly guideline inspection results could be in line with our initial assumption that web portal designers approach the interface design from their perspective, not taking into consideration wishes and opinions of the end users. For this reason, although the evaluation approach has once again shown to be cost-effective (the fewer number of specialists with higher expertise and the employment of simpler evaluation form have provided enough quantitative and qualitative information), raised concerns should be clarified. In future work the issue of “instant” specialists [31] will be considered, although it would be hardly avoided due to the inadequate number of resident HCI experts and the high cost of engaging foreign specialists.

Overall, the results of this comprehensive evaluation study supported the assertion that we should not rely on isolated evaluations. Instead, yet again we concluded that usability assessment methods should be combined, giving rise to different kinds of usability improvement suggestions. Our experience indicated that the scenario-based usability testing is shown to be consistent and that the assessment can be made cost-effective by employing a faster and less expensive procedure which provides stability of measures with reduced sample size. To improve the applicability of the approach to practice and to achieve its broad generalization, our future work will consider the inclusion of a cross-cultural sample. Concerning the conducted heuristic evaluation and its disagreement with the results achieved with user testing, the inspection with “real” HCI experts in usability assessments will be conducted.

Acknowledgements This chapter describes the results of research being carried out within the project 177-0361994-1998 Usability and Adaptivity of Interfaces for Intelligent Authoring Shells funded by the Ministry of Science, Education and Sports of the Republic of Croatia. Experiment was conducted at Department of Visual Communication Design, Arts Academy, University of Split.

References

1. Bevan, N. (1995) Measuring usability as quality of use. *Software Quality Journal* 4: 115–150.
2. Boye, J. (2006) Improving portal usability. *CMS Watch* (available at <http://www.cmswatch.com/Feature/151-Portal-Usability>).
3. Brantley, S., Armstrong, A., and Lewis, K. M. (2006) Usability testing of a customizable library web portal. *College & Research Libraries* 67: 2.
4. Brooke, J. (1996) SUS: a “quick and dirty” usability scale. In: Jordan, P. W., Thomas, B., Weerdmeester, B. A., and McClelland, A. L. (Eds.): *Usability Evaluation in Industry*. Taylor and Francis, London, pp. 189–194.
5. Can, F., Kocberber, S., Baglioglu, O., Kardas, S., Ocalan, H.C., and Uyar, E. (2008) Bilkent news portal: A personalizable system with new event detection and tracking capabilities. *SIGIR'08*, July 20–24, Singapore.
6. Carr, A. (2004) Tapor: A case study of web portal usability (available at http://tapor.ualberta.ca/News/TAPoR_UL.pdf).
7. Carstens, D. S., and Patterson, P. (2005) Usability study of travel websites. *Journal of Usability Studies* 1: 1.
8. GemiusAudience (2008) http://www.valicon.net/uploads/tablica_zaweb_aktualni_podaci_zakolovoz_o8.pdf.
9. GFK Croatia (2008) <http://www.gfk.hr/press1/infopis.htm>.
10. Granić, A., Mitrović, I., and Marangunić, N. (2009) Web portal design: an employment of a range of assessment methods. In: Papadopoulos, G. A., Wojtkowski, W., Wojtkowski, W. G., Wrycza, S., & Zupancic, J. (Eds.): *Information Systems Development: Towards a Service Provision Society*. Springer, New York, NY, pp. 131–139.
11. Granić, A., Mitrović, I., and Marangunić, N. (2008) Experience with usability testing of web portals. In: Cordeiro, J., Filipe, J., and Hammoudi, S. (Eds.) *Proceedings of the 4th International Conference on Web Information Systems and Technologies – WEBIST 2008..* Portugal: INSTICC PRESS, pp. 161–167.
12. Hornbæk, K. (2006) Current practice in measuring usability: Challenges to usability studies and research. *International Journal of Man-Machine Studies* 64: 2.
13. Hornbæk, K., and Law, E. L.-C. (2007) Meta-analysis of correlations among usability measures. *CHI 2007 Proceedings*, April 28–May 3, San Jose, CA, USA.
14. Hvannberg, E., Law, E. L.-C., Larusdottir, M. (2007) Heuristic evaluation: Comparing ways of finding and reporting usability problems. *Interacting with Computers* 19: 225–240.
15. ISO/IEC 25062:2006 (2006) Software engineering – Software product Quality Req. and Evaluation (SQuARE) – Common Industry Format (CIF) for usability test reports.
16. Kellar, M., and Watters, C. (2006) Using web browser interactions to predict task. *WWW 2006*, May 23–26, Edinburgh, Scotland.
17. Klaussegger, C. (2006) Evaluating internet portals – An empirical study of acceptance measurement based on Austrian National Tourist Office’s service portal. *Journal of Quality Assurance in Hospitality & Tourism* 6: 3–4.
18. Lewis, J.R. (2005) Introduction to usability testing. Tutorial given at *HCI International 2005*. July 22–27. Las Vegas, Nevada, Vol.13, No.6, pp. 29–33.
19. Lewis, J. R. (2006) Sample sizes for usability tests: Mostly math, not magic. *Interactions* Nov/Dec.
20. MIT Usability Guidelines (2004) <http://web.mit.edu/is/usability/selected.guidelines.pdf>.
21. Nielsen, J. (1993) *Usability Engineering*. Academic, Boston, pp. 25–62.
22. Nielsen, J. (1994) Heuristic evaluation. In: Nielsen, J., and Mack, R. (Eds.) *Usability Inspection Methods*. John Wiley and Sons, Inc., New York, NY.
23. Telang, R., and Mukhopadhyay, T. (2005) Drivers of Web portal use. *Electronic Commerce Research and Applications* 4: 49–65.
24. Theng, Y. L., and Soh, E. S. (2005) An asian study of healthcare web portals: Implications for healthcare digital libraries. In: *Proceedings of the 8th International Conference on Asian Digital Libraries*, ICADL, Bangkok.

25. Tomić-Koludrović, I., and Petrić, M. (2004) Identities on the net: Gender and national stereotypes on croatian broad-reach portals. *Društvena Istraživanja* 13: 4–5.
26. Tractinsky, N., Katz, A., and Ikar, D. (2000) What is beautiful is usable. *Interacting with Computers* 13: 2.
27. Tsui, W. C., and Paynter, J. (2004) Cultural usability in the globalisation of news portal. In: Masoodian, M., Jones, S., and Rogers, B. (Eds.) *Computer Human Interaction, Proceedings of the 6th Asia Pacific Conference, APCHI*. LNCS 3101. Springer–Verlag, Berlin–Heidelberg.
28. Tullis, T. S., and Stetson, J. N. (2004) A comparison of questionnaires for assessing website usability. *Proceedings of UPA Conference*, Minneapolis, MN. <http://home.comcast.net/~tomtullis/publications/UPA2004TullisStetson.pdf>.
29. Waloszek, G. (2001) Portal usability – Is there such a thing? *SAP Design Guild*, Edition 3. http://www.sapdesignguild.org/editions/edition3/overview_edition3.asp.
30. Wood, J. (2004) Usability heuristics explained. *iQ Content*. http://www.iqcontent.com/publications/features/article_32.
31. Wright, P., and Monk, A. (1991) A cost-effective evaluation method for use by designers. *International Journal of Man-Machine Studies* 35(6): 891–912.

IT Knowledge Requirements Identification in Organizational Networks: Cooperation Between Industrial Organizations and Universities

Peteris Rudzajs and Marite Kirikova

Abstract ICT professionals face rapid technology development, changes in design paradigms, methodologies, approaches, and cooperation patterns. These changes impact relationships between universities that teach ICT disciplines and industrial organizations that develop and use ICT-based products. The required knowledge and skills of university graduates depend mainly on the current industrial situation; therefore the university graduates have to meet industry requirements which are stated at the time point of their graduation, not at the start of their studies. Continuous cooperation between universities and industrial organizations is needed to identify a time and situation-dependent set of knowledge requirements, which lead to situation aware, industry acknowledged, balanced and productive ICT study programs. This chapter proposes information systems solutions supporting cooperation between the university and the industrial organizations with respect to curriculum development in ICT area.

Keywords Educational institution · Knowledge requirements · Study program · Industrial standards

1 Introduction

Educational institution is a member of an educational “ecosystem” [12] that consists of scientific and industrial organizations as well as public/governmental institutions and schools [19]. For the educational institution to be a productive member of the ecosystem it is necessary to satisfy the needs of scientific, industrial, and other organizations. In this chapter we focus on the cooperation between industrial organizations and the university in the context of knowledge provision in the ICT field. This chapter addresses the problem that arises because of rapid developments and changes in ICT area, namely, the problem that industrial requirements stated

P. Rudzajs (✉)

Department of Systems Theory and Design, Riga Technical University, Riga, Latvia
e-mail: peteris.rudzajs@rtu.lv

at a particular time point, when a study program is started, have already changed at the time point when the first students graduate the program. In order to identify, monitor, reflect, and anticipate changes in knowledge requirements for both educational and industrial partners we propose to develop a supporting education – industrial information system (EIIS). This chapter presents the educational knowledge requirements identification part of EIIS architecture. This part of EIIS architecture is designed for handling heterogeneous sources of information that are relevant for continuous knowledge requirements identification and monitoring. The main research question addressed here is how to facilitate requirements amalgamation and representation in a unified form that is relatively easy to maintain and change.

This chapter is structured as follows: In Section 2 the main problems of knowledge requirements identification are characterized, the proposed solution outlined and related works briefly overviewed. In Section 3 the issue of knowledge source and representation heterogeneity is addressed. The proposed modes of information handling are described and exemplified in Section 4. Section 5 consists of conclusions and directions of future work.

2 Problems, Proposed Solution, and Related Works

ICT field is the educational area that faces continuous changes in industrial requirements of the knowledge possessed by university graduates. Those changes are so rapid that frequently the industrial requirements that were taken into consideration by universities at the time point of students starting a particular study program are not valid anymore at the time point of graduation. In addition, industrial representatives due to the necessity to focus on the utilization of advanced technologies do not always estimate correctly the value of basic knowledge (e.g., physics and mathematics, systems theory) that contribute to abstract and systems thinking abilities of students [1, 8]. Small and medium companies are not always able to follow advances in ICT field development and therefore cannot state realistic requirements for their future employees' knowledge; and, due to their not fully advanced knowledge in the field, do not trust university educators who are more focused on future trends than on current situation. One of the reasons underlying this problem is the lack of transparent knowledge development trend representations that could be utilized by both university and industrial partners for gaining mutual agreements and providing maximum of support to one another in knowledge provision for ICT students. The purpose of research described in this chapter is to develop gradually required knowledge representation and monitoring system that could provide educational institutions and industrial partners with a transparent view on knowledge (skills, competencies) requirements in the field of ICT development and use. Such system could be a core of EIIS continuously supporting university and industry collaboration.

There are two essential challenges that affect the possibility to develop the above-mentioned system, namely (1) diversity, conceptual heterogeneity, and wide

distribution of knowledge sources for requirements identification and (2) frequent changes in contents of identified sources. The architecture of an intended EIIS therefore has to address these problems by providing multiple ways of information gathering, fusion, and representation. The part of EIIS architecture that addresses these problems is presented in Fig. 1. The figure shows a three-layer architecture where the central element is a knowledge external representation and monitoring service (10) that is intended to provide a transparent representation on knowledge trends in ICT area. This service is supported by an internal unified knowledge representation model/repository (8) that maintains not only the current knowledge representation, but also the history of representations (9). This model is structured in three main knowledge subsystems where each has a different frequency of changes. Knowledge representation should be made on the basis of some existing skills frameworks that define the structure of skills (categories, subcategories, or the level of skills). Some examples include Skills Framework for the Information Age (SFIA) [18], European e-Competence Framework [3], and others.

Changes into the unified internal knowledge representation model are requested by the knowledge requirements identification service SKRq (7). This service, in turn, is supported by several knowledge identification and change services (SKCI) information about required/obtainable/obtained knowledge could be retrieved from (1) employers' published vacancies, (2) register of national occupational standards,

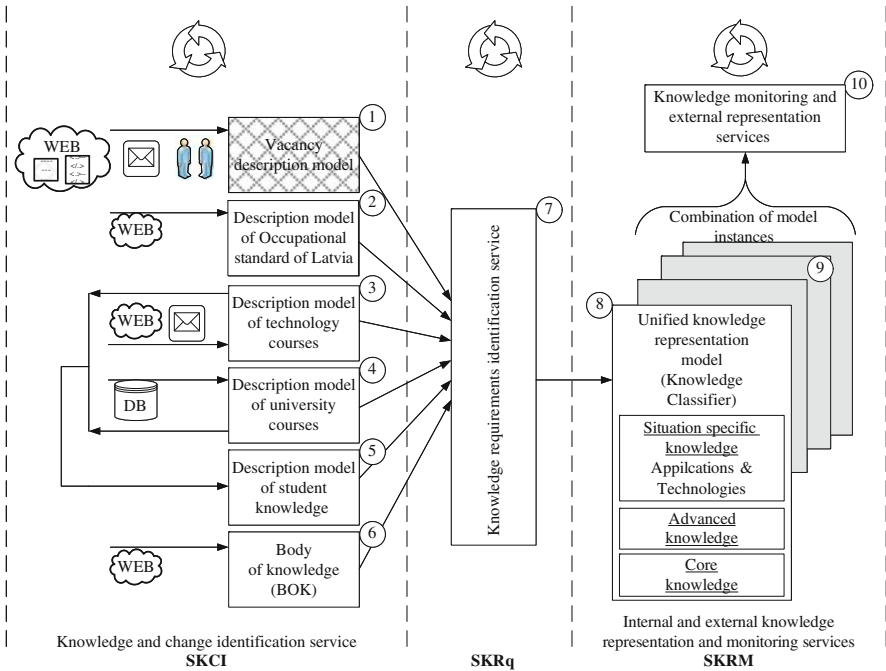


Fig. 1 Part of EIIS architecture

(3) descriptions of industrial certification (technology-oriented) courses, (4) descriptions of university courses, (5) descriptions of student knowledge, (6) descriptions of the so-called Body of Knowledge (BOK) standards for education (both academic and professional) such as Business Analyst BOK, Software Engineer BOK, Project Management BOK [5, 10, 11, 15]. SKCI level supports SKRq level, which, in turn, supports internal and external knowledge representation and monitoring services (SKRM). Thus, the system is “fed” by SKCI. The number of these services is not limited to the ones presented in Fig. 1. Two of the presented services (3 and 4) utilize internal systems feedbacks additionally to the investigation of external environment. Multiple knowledge identification services are used due to the heterogeneity of information sources. Each service provides several modes of operation (starting from manual to fully automated ones) that allows flexible customization of information acquisition depending on the acquisition purpose and availability of source knowledge.

The development of EIIS, in general, and services described in this chapter, in particular, are based on related work in organizational ecosystems [7, 12], intelligent agents [24], information fusion [22], ontology matching and maturing [2, 4], as well as knowledge mapping in the area of ICT [14, 21], and business intelligence [23].

3 Diversity of Knowledge Requirements Sources

Every previously identified source has its own form of knowledge representation – usually free-formed description of information published in web or sent to educational institution by e-mail. Other information forms include databases, annotated texts [20], and conversations with the employers. Each SKCI level service can provide information source searching facilities, information change searching facilities (for sources already found), and information retrieval facilities in fully automated, semi-automated and manual modes. Information changes searching web crawler is developed [17] and used as a part of any SKCI level service. Methods of information handling that utilize the crawler together with manual operations are described in Section 4. Information sources of knowledge requirements and forms of information relevant for invoking particular SKCI corresponding to the type of information source are summarized in Table 1.

As mentioned before, the forms of information in knowledge requirement sources are different. The first step to start using information about knowledge requirements is to retrieve and transform it. To structure the retrieval and transformation efforts, the initial general source sensitive description models were developed. They are represented in Figs 2, 3, 4, 5, and 6. We can see that these models differ widely one from another. Therefore, SKRq is needed to handle these differences and prepare information for inclusion in the unified knowledge representation model (SKRM layer in Fig. 1).

As proposed in the information system architecture of knowledge monitoring, one of the information sources is vacancy descriptions (element 1 in Fig. 1).

The largest variety of forms is found in employers’ published vacancies. These forms include (1) annotated text, (2) free-form text, (3) e-mail, and (4) conversations

Table 1 Information sources of knowledge and their forms

Information source/form	Vacancy description	Occupational standard	Technology course description	University course description	Student knowledge description	BOK
1. Annotated text	+			+		
2. Free-form text	+	+	+	+		+
3. E-mail	+		+	+	+	
4. Conversation	+			+	+	
5. Database				+	+	

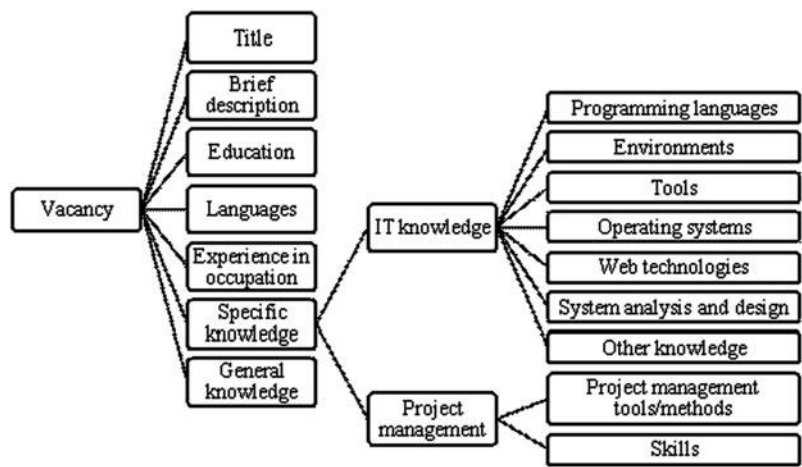


Fig. 2 Description model of a vacancy

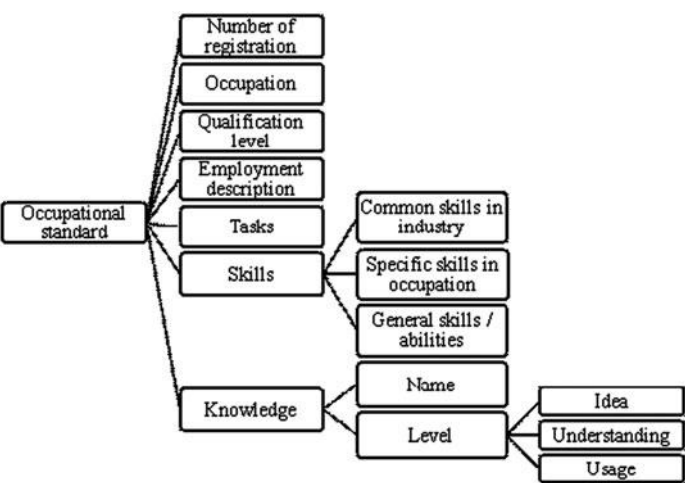


Fig. 3 Description model of occupation in occupational standard

Fig. 4 Description model of technology course/exam

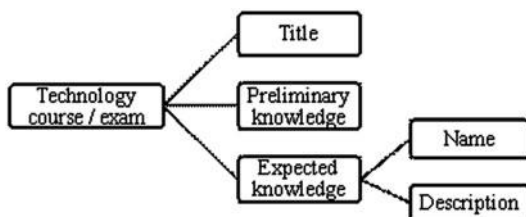


Fig. 5 Description model of university course

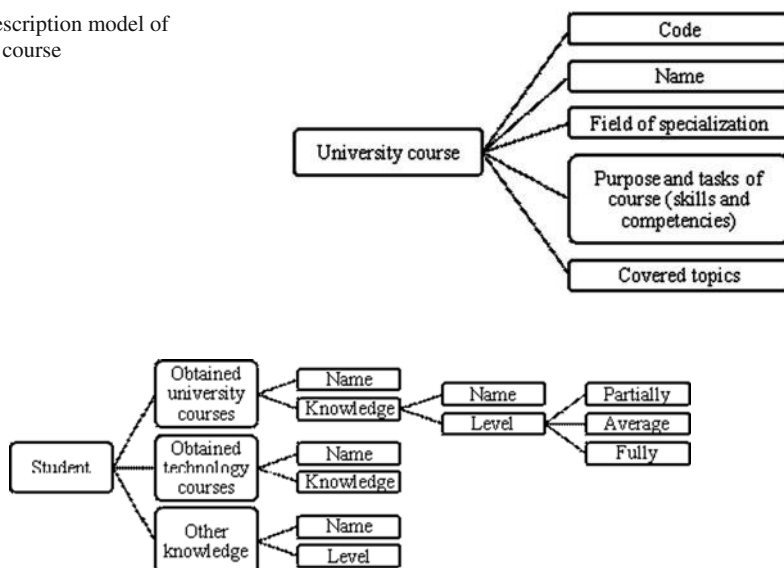


Fig. 6 Description model of student knowledge

with the employers (see column 1 in Table 1). To make clear the conceptual structure of vacancy contents published by employers, simplified description model (see Fig. 2) is constructed. In the published vacancies we usually can find such conceptual parts as follows: (1) title of vacancy (occupation), (2) brief description of vacancy followed by knowledge requirements. (3) requirements of education, (4) requirements of knowledge in languages, (5) experience in occupation requirements, (6) specific knowledge requirements (this describes required knowledge of various programming languages, environments, tools, operating systems, web technologies, system analysis and design, project management skills, etc.), and (7) general knowledge requirements. In Fig. 2, specific knowledge is described in more detail than usually it is available in vacancy descriptions.

National occupational standards (relevant for element 2 in Fig. 1) are usually published in a structured manner, i.e., particular information units can be emphasized in the description of occupation: number of registration, occupation title, qualification level, employment description, tasks, skills (classified as common skills in

industry, specific skills in occupation, and general skills/abilities) and knowledge levels – idea, understanding, and usage – are specified [16]. Conceptual model of occupational standard description is given in Fig. 3. For example, in the registry of occupational standard of Latvia [16], there are described (standardized) various occupations in the field of computer science and information technology. They are programmer, software engineer, computer system and computer networks administrator, system analyst, computer system technician, and information technology project manager. These standards are freely accessible online [16].

The purpose of technology courses (relevant for element 3 in Fig. 1) is to certify a person in the area of some technology, e.g., Microsoft offers certification exam “Designing, Assessing, and Optimizing Software Asset Management (SAM)” [13]. After the completion of this exam, Microsoft issues a certificate that acknowledges the acquired knowledge and skills. The descriptions of exams are publicly available [9, 13]. The description defines required preliminary knowledge and expected knowledge (in the form of the name of knowledge elements and their description) after the exam has been taken. After analyzing certification courses and exams offered by IBM and Microsoft, a conceptual model of their description was constructed (see Fig. 4).

In an educational institution every course (relevant for Element 4 in Fig. 1) is planned by defining various attributes, such as course name, field of specialization, purpose and tasks of the course in the form of skills and competencies, and the description of topics covered by the course. Conceptual model of course description is given in Fig. 5.

By taking university courses, the student of a higher education institution accumulates knowledge. Therefore, university courses could serve as the basis for student knowledge identification. By combining courses we could get the “ideal” set of student’s accumulated knowledge. In the ideal case – the student has fully obtained knowledge from his courses. The prerequisite to map student knowledge to unified knowledge representation model is to have university courses mapped to unified knowledge representation model. In the next step, courses taken by student should be identified by describing the level of obtained knowledge units (measuring as – partially, average, and fully). This ensures the evaluation of real knowledge obtained by the student. The conceptual model of student knowledge is given in Fig. 6.

To complete student knowledge we should use the results of knowledge represented by university and technology courses. In this case, the knowledge obtained from the courses is automatically assigned to the student. For assignment of courses taken by student, the functionality of SKRq can be used.

BOK standards are defined by the relevant professional association. This leads to differences in the structure of these standards. We propose to identify the name, developer association, version, and the year of publication of every BOK standard. These standards are significant both for educational institutions and for employers in their industry, because they define best practices in various occupations. Therefore, a periodical review and analysis of these standards are vital for the development of educational knowledge requirements.

4 Customized Handling of Knowledge Requirements Sources

In order to retrieve and analyze knowledge contents of identified knowledge sources it is necessary to take into consideration the form of information representation (rows in Table 1 and columns in Table 2). Information-handling methods depend also on the type of knowledge sources represented in columns of Table 1. In this section we discuss information-handling methods for three sources. The method of vacancy description handling is the most complicated and therefore is described in detail as an example for better understanding of other methods. Several other information handling methods are still under the development and are not considered in this section in detail.

4.1 Method for Vacancy Description Handling

To get information about vacancies, appropriate methods and technologies should be used for every form of information source (Tables 1 and 2):

1. Retrieval of *annotated vacancy* description is possible with the use of web agent. In this case web agent (crawler) searches the web, identifying links to vacancy descriptions. Identified annotated descriptions of vacancies can be presented by different document models. If the annotation [20] published by the employer would be built by using unified knowledge representation model, transformation could be done semi-automatically. If the annotation is structured differently, then the published model initially should be mapped manually, and next time when a similar model is retrieved, the mapping process may be done automatically on the basis of the initial mapping between models.
2. Web agents are useful when *free text* form vacancy description has to be retrieved. After retrieving vacancy description, the contents need to be structured, i.e., description should be mapped to unified knowledge requirement model manually.
3. If a vacancy description is sent via e-mail, the content should be mapped to unified knowledge requirement model.
4. In conversations with employers useful information could be captured, e.g., the employer describes future needs for specialists and gives their description defining knowledge and skills expected after some period of time. This type of conversations is important for educational institutions as the trends in the labor market could be noticed to deliver particular optional courses in order to prepare students for the chosen vacancies and satisfy market demand.

In cases 2–4 definition of SKRq is required (element 7 in Fig. 1). The main role of this service is to help to structure and to map identified knowledge to unified knowledge representation model. This service provides automatic analysis of the terms in the vacancy description. On the basis of the analysis of knowledge requirements,

Table 2 Vacancy retrieval process depending on information representation form

1. Annotated text	2. Free-form text	3. E-mail	4. Conversation
The need for specialist			Future need for specialist
Employer defines knowledge requirements for specialist			
Employer produces vacancy description including required knowledge and skills			
Employer publishes vacancy description in free-form text with <i>annotation</i>	Employer publishes vacancy description in free text form	Employer sends vacancy using e-mail	In the form of conversation employer describes future needs for vacancies and gives their description
Web agent finds published vacancy by searching the web			
Web agent informs users who are responsible for employers' database about the vacancy found			
There was made mapping for annotated text to unified knowledge representation model	Vacancy description is reviewed		
Vacancy description is automatically transformed to unified knowledge representation model	Acquired knowledge about future vacancies manually is transferred to unified knowledge representation model by extracting required knowledge, skills, and abilities. This could be done by using knowledge identification service		Acquired knowledge about future vacancies is transferred manually to unified knowledge representation model by specifying the time when vacancy could open. Knowledge identification service can be used
	Knowledge identification tool looks for "known" terms identifying knowledge requirements in vacancy description. These suggestions support manual description analysis		
Submit a knowledge requirements to unified knowledge representation model			

recommendations are prepared. For example, web agent has retrieved a vacancy description; the next step is to process it manually (map to unified knowledge requirement model). To automate this process we can use knowledge requirements identification service – vacancy description is analyzed for known terms describing knowledge requirements. These terms are suggested to the user to ease the process of information rereading and the structuring of knowledge requirements. To implement these suggestions, initial dictionary (ontology) of terms used in vacancy descriptions should be engineered.

Vacancy retrieval process depending on information representation form is exemplified in Table 2.

It is necessary to note that multilingual [6] information identification and retrieval services are needed because of global nature of IT labor market.

4.2 Method for Occupational Standards Handling

For handling occupational standards we propose periodical browsing of registry of occupational standards by crawler and identifying changes in desired (e.g., system analyst) occupation descriptions. This is the way for ensuring up-to-date information about knowledge and skills required in a certain occupation. Standard change frequency is low, e.g., most of the standards [16] were last time changed in 2002 and 2003. Web agent searches the registry, identifying new standards or identifying changes in existing ones. If changes are identified, standard review is required to determine what exactly has changed. In this step knowledge identification service could be used. Knowledge units in standard description should be identified and mapped to unified knowledge representation model.

4.3 Method for Technology Course Description Handling

After analyzing defined knowledge units in the descriptions of technology courses and exams, two types of information retrieval were identified – (1) search the web and (2) receive course lists and their descriptions from some collaborative institution (industry partner).

If course descriptions are published in web, they can be retrieved using web agent technology – search the web, identify courses, and represent search results to a user. In the search results we could identify three cases: (1) new courses are discovered, (2) changes in existing courses are identified comparing with previous search results, and (3) some courses could not be found anymore. In the first case, by using knowledge identification service, mapping to unified knowledge representation model should be done. In the second case, selection of results from previous mapping results and remapping the changed ones (by identifying new knowledge in existing courses) should be done. If the conditions for the third case are true, then the course should be deleted from the database keeping existing mapping of

terms. These mappings could be used for other course descriptions to map new knowledge.

When the educational institution receives e-mails from some collaborative partners, the flow of events is similar to the above mentioned only with one exception – web agents are not used.

5 Conclusions

This chapter presents a part of architecture of EIIS that is developed with the purpose to establish continuous collaboration between the industry and university with respect to transparent and well-motivated student knowledge development strategies in the area of ITC education. The sub-architecture of EIIS consisting of three layers (knowledge and change identification, knowledge requirements identification, and internal and external knowledge representation and monitoring) are proposed and described. This chapter focuses mainly on the first layer of the architecture by describing a variety of knowledge sources and corresponding information handling methods. The web agent/crawler that is used in several steps of the methods has been developed and tested. First attempts for organizing unified representation model have been made; however, further research is needed to achieve minimization of manual comparison of knowledge structures. Mapping of all description models into unified knowledge representation model would give an opportunity to analyze different aspects of student knowledge and automatically or semi-automatically [19] obtain information regarding the following indicators:

- correspondence of student knowledge to occupational standards,
- correspondence of student knowledge to different occupations,
- correspondence between knowledge required by vacancies and knowledge provided by study courses (mandatory and elective),
- correspondence of university courses and industrial certification courses, etc.

Research work presented in this chapter revealed that ITC organizations differ considerably with respect to detail of their internal job descriptions and in many cases their wishes change when explicit knowledge about university knowledge development is presented. They see the EIIS as a tool for workspace acquisition and analysis of labor market knowledge requirements. Most probably each component of the system will have to be used in all three modes (manual, semi-automated, fully automated modes), depending on the organizations that use EIIS and available information sources of knowledge requirements. Nevertheless, one of the main directions of future work is introduction of more sophisticated ontology matching and information fusion methods. Another direction of future research is a more formal utilization of feedback mechanisms in the university – industry ecosystem in general and in EIIS in particular.

References

1. Armoni, M., and Gal-Ezer, J. (2006) *Reduction – An abstract thinking pattern: The case of the computational models course*. In: *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education (SIGCSE'06)*, Houston, USA, March 1–5, pp. 389–393.
2. Braun, S., Schmidt, A., and Walter, A. (2007) *Ontology Maturing: A collaborative Web 2.0 Approach to Ontology Engineering*, Workshop on Social and Collaborative Construction of Structured Knowledge at WWW.
3. European e-Competence Framework, Retrieved June 29, 2009, from: <http://www.ecompetences.eu>.
4. Euzenat, J., and Shvaiko, P. (2007) *Ontology Matching*, Springer, New York.
5. Fanning, F., and Camplin, J. C. (2008) *Body of Knowledge*. Professional Safety 53(11): 54.
6. Gautschi, H. (2005) *Search in Any Language*. EContent 28(5): 29.
7. Ghose, A., and Koliadis, G. (2008) *Actor Eco-systems: Modeling and Configuring Virtual Enterprises*. In: *Congress on Services Part II*, Beijing, China, September 23–26, pp. 125–132.
8. Hazzan, O., and Tomayko, J. E. (2005) *Reflection and Abstraction in Learning Software Engineering's Human Aspects*. Computer 38(6): 39–45.
9. IBM (2009) *IBM Professional Certification Program*. Retrieved May 1, 2009, from: <http://www-03.ibm.com/certify/index.shtml>.
10. Institute of Electrical and Electronics Engineers (2004) *Guide to the Software Engineering Body of Knowledge*. Retrieved May 5, 2009, from <http://www.computer.org/portal/web/swebok>.
11. International Institute of Business Analysis (2006) *A Guide to the Business Analysis Body of Knowledge*. Retrieved May 5, 2009, from http://www.theiiba.org/AM/Template.cfm?Section=Version_1_6.
12. Kirikova, M., Grundspenkis, J., and Sukovskis, U. (2008) *Educational “ecosystem” for information systems engineering*. In: I. Horvath, and Z. Rusak (Eds.) *Proceedings of the 7th International Symposium on Tools and Methods of Competitive Engineering (TMCE)*, Turkey, April 21–25. Delft University of Technology, pp. 769–783.
13. Microsoft learning (2009) *Description of Microsoft Certification exam “Designing, Assessing, and Optimizing Software Asset Management (SAM)”*. Retrieved May 5, 2009, from: <http://www.microsoft.com/learning/en/us/Exams/70-673.aspx>.
14. O*NET Resource Center (2009) *About O*NET*, Retrieved May 2, 2009, from: <http://www.onetcenter.org/overview.html>.
15. Project Management Institute (2004) *A Guide to the Project Management Body of Knowledge*. Retrieved May 5, 2009, from <http://www.pmi.org/PMBOK-Guide-and-Standards.aspx>.
16. Registry of Occupational standard of Latvia. Available at: <http://www.izmpic.gov.lv/PSR/psr.html>.
17. Rudzajs, P. (2008) *Development of knowledge renewal service for the maintainance of employers' database*, Bachelor thesis (Riga Technical University), Riga, Latvia.
18. SFIA Foundation (2009) *Skills Framework for the Information Age (SFIA)*, Retrieved June 29, 2009, from: <http://www.sfia.org.uk>.
19. Strazdina, R., Stecjuka, J., Andersone, I., Kirikova, M. (2008) *Statistical analysis for supporting inter-institutional knowledge flows in the context of educational system*, Accepted at the 19th International Conference on Information Systems development (ISD2008), Paphos, Cyprus, August 25–27.
20. Studer, R., Grimm, S., and Abecker, A. (2007) *Semantic Web Services. Concepts, Technologies, and Applications*. Springer, Berlin.
21. Subrahmanyam, G. (2009) *A dynamic framework for software engineering education curriculum to reduce the gap between the software organizations and software educational institutions*. In: *Proceedings of the 2009 22nd Conference on Software Engineering Education and Training*, Washington, February 17–20, pp. 248–254.

22. Torra, V. (2003) *Information Fusion in Data Mining*. Springer, Berlin.
23. Viaene, S. (2008) *Linking Business Intelligence into Your Business*. IT Professional 10(6): 28–34.
24. Zhong, N., Liu, J., and Yao, Y. (2003) *Web Intelligence*. Springer, Berlin.

A Knowledge Tree Model and Its Application for Continuous Management Improvement

Yun Lu, Zhen-Qiang Bao, Yu-Qin Zhao, Yan Wang, and Gui-Jun Wang

Abstract This chapter analyzes the relationship of organizational knowledge and brings forward that organizational knowledge consists of three layers: core knowledge, structural knowledge, and implicit knowledge. According to the principle of knowledge maps, a dynamic management model of organizational knowledge based on knowledge tree is introduced and the definition of the value of knowledge node is given so that the quantitative management on knowledge is realized, which lays a foundation for performance evaluation of knowledge management. We also carefully study the application of knowledge tree in service quality management of hospital organizations and management innovation process and give the example of cooperation in endoscopic surgery to establish a knowledge tree about operational cooperation degree, which states the principle of organizational knowledge management and the knowledge innovation process of continuous management improvement.

Keywords Knowledge management · Knowledge tree · Value of knowledge · Service management

1 Introduction

For enterprises or organizations, the changes in environment in competition bring about changes in organization management; knowledge becomes the core asset which produces social value and economic value [12]. Many conceptions and ideas of knowledge management were raised by management scholars, such as knowledge maps, team learning, and organizational memory [5, 7, 11]. However, on reviewing relevant research results on knowledge management, we find a few limitations [4]: (1) relevant researches on knowledge management remain in the qualitative phase and little quantitative work has been reported, and they cannot provide evidence for

Y. Lu (✉)
Subei People's Hospital, College of Clinical Medicine, Yangzhou University, Jiangsu 225001,
P.R. China
e-mail: yzbzq@163.com

the operation of organization management; (2) knowledge flows, information flows, and value flows reflect the operational process of an enterprise from three views, but those research results cannot provide a model with which we can study from the different views simultaneously; (3) because of the characteristics of knowledge, the problem of how to measure knowledge hinders the research work on knowledge management and so we cannot find a quantitative method to measure the knowledge achievement, which meets the requirements of knowledge management [10, 13].

With the development of social economy and information, quality of service management is increasingly becoming the focus of public's attention, as well as theory cycle. This chapter focuses on the process of organizational knowledge structure and knowledge transformation. A knowledge tree model for continuous management improvement is put forward, which is driven by the value goal of an enterprise, and we also carefully study the application of the knowledge tree in service quality management of hospital organizations and management innovation process. We take the cooperation in endoscopic surgery as an example to establish a knowledge tree about operational cooperation degree, which expresses the principle of organizational knowledge management and the knowledge innovation process of continuous management improvement.

2 Structure of Organizational Knowledge

The relation of organizational knowledge is very complex. But we can select a view carefully and analyze the structure. In the view of psychology, the structure of organizational knowledge consists of three layers: core knowledge, structural knowledge, and implicit knowledge.

2.1 Core Knowledge

In an organization, there is a part in its knowledge structure which describes its mission, value direction, and basic goal. This part is the core layer of organizational knowledge structure, which provides value measure in an enterprise to scale behaviors of the organization and its members and guides them to develop and accumulate organizational knowledge. The essential goal should be to create value. Different enterprises existing in different environments take different value goals. For example, the cooperation degree between doctors and theatre nurses in whole operations is the basic criterion which weighs the work quality of operating rooms in hospitals.

2.2 Structural Knowledge

Structural knowledge is the knowledge about sub-goal or means and behaviors to realize the sub-goal. With structure knowledge we can connect value goal in core

layer with the sub-goal, means, and behaviors. These causal connections express paths reaching the value goal of an organization, which are called knowledge chains. All the knowledge chains constitute a tree, with the root expressing the value goal of an organization. A knowledge tree not only expresses a knowledge structure, but also shows the dynamic process in which the organization and its members adapt themselves to their environment and create and accumulate new knowledge, so as to achieve their goal. Structural knowledge exists in the assets of a hospital, such as equipment, instrument, computer software, and also in the hospital's organization management system, such as institutes, technical standard, and operational regulations. They usually take on certain structural forms, which are explicit knowledge in organizations and can be shared with organizational members [9].

2.3 Implicit Knowledge

Implicit knowledge is the knowledge, experiences, and skills held by the individuals of an organization, which relies on human brains and flows with the organizational members, such as health professionals' experiences and skills. When a member of an organization leaves the organization, the implicit knowledge which he (or she) holds will leave the organization at the same time. Structural knowledge owned by organizations is different from implicit knowledge as it remains in organizations. However, implicit knowledge is the resource of structural knowledge. The process of expansion of organizational knowledge structure is one in which implicit knowledge is transformed into structural knowledge. One of the tasks of knowledge management is to provide appropriate environment and let organizational members develop their implicit knowledge. On the other hand, all managers of a firm should dig into the implicit layer of knowledge structure and take great advantage of the implicit knowledge and transform it into structural knowledge [3, 8].

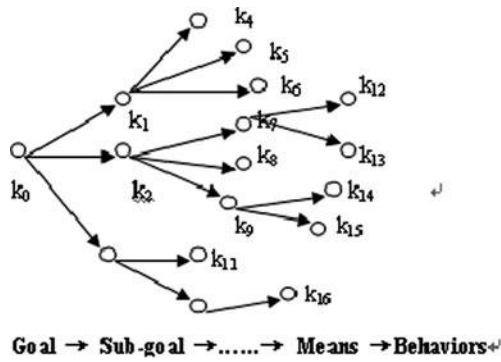
3 Knowledge Tree Model

In nature, knowledge tree is a hierarchic knowledge map, which expresses the causal or subordinate relationship of related organizational knowledge that helps to realize the organizational goal. For the characteristics of symmetry, good figure, continuity, and hierarchy, knowledge map is widely used [1, 2]. Knowledge tree model develops more management functions based on hierarchic knowledge map.

3.1 Description of Knowledge Tree

Consider a hierarchic knowledge structure for an organizational goal. The structure is a set K of knowledge nodes which are related. All the relations r comprise a relation set R . Knowledge node set K and knowledge relation set R comprise a

Fig. 1 Structure of knowledge tree



knowledge tree $D=(K,R)$. To study knowledge tree further, several conceptions are introduced as follows:

Parent node is the one representing goal knowledge of two conjoint knowledge nodes. In Fig. 1 node k_0 is the parent node of nodes k_1, k_2, k_3 .

Child node is the one representing the means for the goal of two conjoint knowledge nodes. In Fig. 1, k_4, k_5, k_6 are children nodes of k_1 .

Relation is conjoint mode of two nodes. We represent relation of node k_i and k_j by r_{ij} . Relation r has two types:

- (1) Combination means that all children nodes of parent nodes perform the goal of the parent node together or are sub-goals for the goal, marked with *and* or $\vee$.
- (2) Replacement means that all children nodes are replaceable means or plans to realize the goal of their parent node, marked with *or* or $\wedge$.

Root node is the knowledge node without parent node. There is only one root node in a knowledge tree.

Leaf node is a knowledge node without child node.

Knowledge chain is a node series from a knowledge node to one of its reachable leaf nodes. For instance, in the knowledge tree (Fig. 1) (k_0, k_1, k_4) is another knowledge chain of k_0 .

Knowledge sub-tree of a knowledge node is constituted by all knowledge chains of the knowledge node. For example, in Fig. 1 $(k_2, k_7, k_8, k_9, k_{12}, k_{13}, k_{14}, k_{15})$ is a knowledge sub-tree of node k_2 .

3.2 Value of Knowledge Tree

Definition The contribution to the organizational goal of a knowledge node in a certain time is called the value of the knowledge node.

The definition of value provides a measure for knowledge. For convenience we make two assumptions as follows:

- (1) Value of a leaf node is equal to its contribution to the organizational goal.
- (2) Value of any other node is realized by its children nodes.

According to the definition and assumption above, we know that the values of all knowledge nodes are realized by their leaf nodes in the knowledge tree.

Suppose the value of node i is V_i and its children nodes are $k_{ij}(j = 1, 2, \dots, n)$. Their values are $v_{ij}(j = 1, 2, \dots, n)$, then

For relation of combination

$$v_i = \sum_{j=1}^n v_{ij}$$

For relation of replacement

$$v_i = \max_j(v_{ij})$$

3.3 Description of Knowledge Node

Knowledge node is the basic unit which constitutes the knowledge tree; every node is one section of the knowledge chain aimed to ensure the realization of organizational goals. Therefore, management of knowledge node is the brief content of operations management of organizational knowledge. The basic attributes of a knowledge node are listed in Table 1.

In Table 1, according to the demand of management, knowledge is divided into categories such as system (S), organization (O), people (P), technique and design (T), management (M), and environment (E).

Table 1 Attributes of a knowledge node

Name	Meaning of attribute
N_code	Code of a node
N_name	Name of a node
Type	Type of knowledge
Department	Department keeping knowledge
P_code	Code of parent node
Relation	Relation with parent node
Value	Value of knowledge node
S_time	Statistic time of value
Innovator	Organization or member of innovating the node
I_time	Last innovatory time of node

4 Knowledge Tree Model on Operational Cooperation Degree

We apply the knowledge tree model to quantifiable performance management system of quality control in a hospital and establish the performance structure of quality management which has been used to develop management information system. Due to limitations on length, this chapter provides only the cooperation quality of endoscopic surgery as an example to discuss the use of knowledge tree model in the operating room. We have built a knowledge tree for continuous management improvement of operational cooperation degree which is one of the sub-trees of quality management knowledge tree in hospital.

4.1 The Establishment of Knowledge Tree Model of Cooperation Degree in Operations

Cooperating with doctors of surgery to ensure the success of the operation is the essential duty of theatre nurses. The cooperation degree is a key indicator to measure their working quality. Therefore, to note doctors’ degree of satisfaction of every operation, flaws which appear in the operation and suggestions given by the doctor to solve the problems, we designed a questionnaire (Table 2) as follows: (1) degree of satisfaction is divided into three different levels: satisfied (1 point), basically satisfied (0.5 point), dissatisfied (−1 point); (2) flaws given by doctors explain the aspects that need to be improved; (3) suggestions are directions on how to handle matters. According to the result of the investigation, flaws in an operating room are discovered and analyzed in order to extract the implicit knowledge of experts (doctors of surgery) and transform it into explicit knowledge by improving institutions and technologies.

Based on the causal relation between classification of endoscopic surgery and flaws existing in operational cooperation, we establish a knowledge tree model of operational cooperation degree of endoscopic surgery. The relation of nodes is listed in Fig. 2.

Table 2 Questionnaire of cooperation degree in operation

Operation date	Operation name	Doctor of surgery	Degree of satisfaction			Flaws	Suggestion
			Satisfied (1 Point)	Basically satisfied (0.5 Point)	Dissatisfied (−1 Point)		
08. 04. 05	Infertility	Huangyongsheng	✓				
08. 04. 12	Meniscectomy	Shixiaoming			✓	Lighting not properly coordinated	Endoscopic ancillary packaging
08. 04. 30	Cholecystomy	Tanjingwang		✓		Sonoca with the unskilled	Training in the use of ultrasonic scalpel

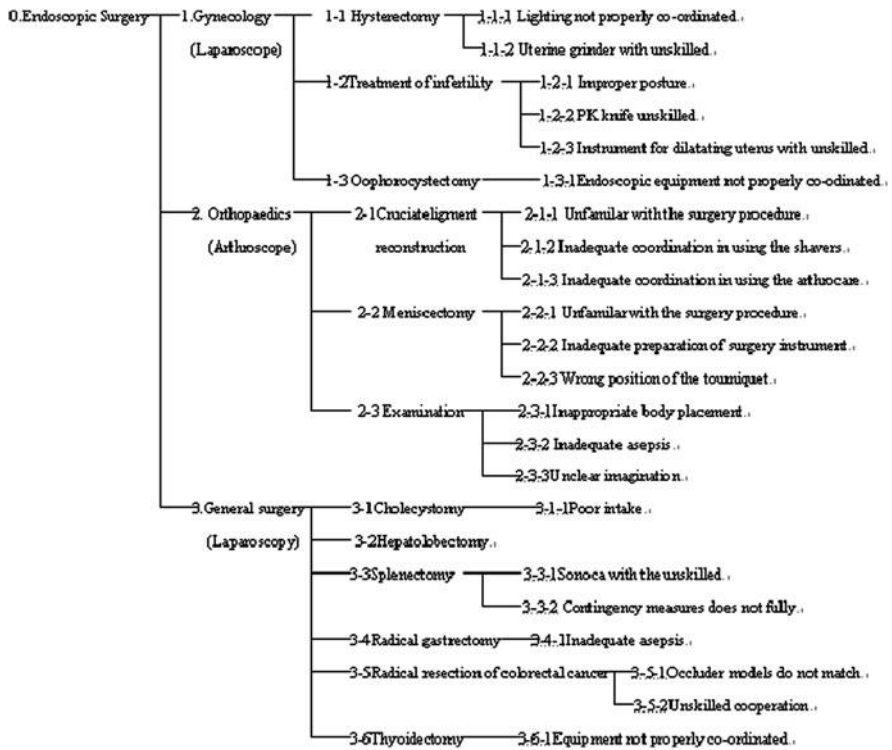


Fig. 2 Knowledge tree model about degree of satisfaction in operation

If the degree of satisfaction of an operation is $S^{(k)}$, the value of a node in the operation is

$$v = \sum_{k=1}^m s^{(k)}$$

where m is the count of operations in plan cycle. If the value of the operational node is v_{ij} , the value of this kind operational node is

$$v_i = \sum_{j=1}^n v_{ij}$$

where n is the kinds of operations. Therefore, the value of the upstream node is the sum of the values of its child nodes.

For comparability, the concept of cooperation degree is introduced. If the count of nodes in knowledge tree is k , the value of the relative node is $v(k)$, and the

count of operations in plan cycle is $N(k)$, then the cooperation degree of this node is

$$f(k) = \frac{v(k)}{N(k)}$$

Table 3 lists the node values of the operation cooperation knowledge tree (including the number of operations, the node value, and the cooperation degree), given by the operation quality management system of Subei People's Hospital at the end of June 2008. To illustrate the management principles of continuous management improvement, Table 3 also gives the relative cooperation degree at the end of December 2008.

4.2 The Continuous Improving Management Principles

Principle 1: Because of the progressive natures of environmental change, the foundational structure of an organization knowledge tree will be maintained quite stable, and we should put emphases of knowledge innovation upon the downstream nodes (leaf nodes). According to this principle, we analyze the flaws of operations first and then draw up improvement measures and transform them into institutions and standardization, which is the knowledge innovation. For example, (1) in response to one case in which bleeding could not be controlled in a splenectomy and an emergency laparotomy equipment was not set up to stop it, which delayed the operation, a new emergency rescue training program about splenectomy is set up. (2) Because the anastomat used in radical resection for the treatment of rectal cancer does not match, we sum up and train on the using convention of anastomat required for intestinal surgery and closer.

Principle 2: Those knowledge nodes and their knowledge sub-trees in which the difference between the finished value and the goal value is greater are the emphases of organizational knowledge management. For example, the degree of satisfaction of operational cooperation is low, because theatre nurses are not familiar with surgical procedures of arthroscopic surgery. Therefore doctors are asked to develop a course on the surgical steps and main points of cooperation and explain by organizing a scene to watch surgical procedures.

Principle 3: When a knowledge tree or sub-tree remains without any variation for a long time and the value of the nodes keep falling, that means a degeneration of the tree or sub-tree and people should work more with an eye on the upstream nodes. The degradation of knowledge tree usually results from common questions of leaf nodes which root from upstream nodes, so we should pay attention to upstream nodes to look for measures when we encounter these problems. For example, we set up an indicate system which is a branch and sub-set of the endoscopic equipment and realize the set management of equipment which greatly simplifies management procedures, improves efficiency, and reduces endoscopic surgery conflicts and disputes, in order to solve the set management problems of endoscope which often

Table 3 Node of endoscopic surgery and value of knowledge

Node number	Node name	Joint knowledge/ example of measures	Operation count	Value of node	Cooperation degree	Ratio of cooperation degree
0	Endoscopic surgery	1. The marking system of endoscopic equipment division and sub-sets 2. Specialist nursing duties 3. Reasonable scheduling and optimizing staffing	374	323	0.86	0.93
1	Gynecology (Laparoscopy)	Manipulation regulation of PK knife	129	118	0.91	0.93
1-1	Hysterectomy	Manipulation regulation of uterine grinder	8	6	0.75	0.80
1-2	Treatment of infertility	1. Manipulation regulation of dilating uterus 2. Surgery position principle	37	32	0.86	0.89
1-3	Adnexectomy	1. Manipulation regulation of the arthrocare instrument	84	80	0.95	0.96
2	Orthopedics	2. Manipulation regulation of the shavers 3. Training program	59	40	0.68	0.84
2-1	Cruciate ligament	Manipulation regulation of the tourniquet	15	9	0.60	0.81
2-2	Meniscectomy		8	5	0.63	0.79

Table 3 (continued)

Node number	Node name	Joint knowledge/ example of measures	Operation count	Value of node	Cooperation degree	Ratio of cooperation degree
2-3	Examination	1. Aseptic manipulation principle	36	26	0.72	0.85
3	General surgery (Laparoscopy)	2. Surgery position principle				
		3. Sonoca operating specifications	186	165	0.89	0.95
3-1	Cholecystomy		137	130	0.95	0.97
3-2	Hepatolobectomy		6	4.5	0.75	0.85
3-3	Splenectomy	Emergency rescue training program	7	5.5	0.79	0.89
3-4	Radical resection of colorectal cancer	Training program	16	11	0.69	0.87
3-5	Radical gastrectomy	1. Tumor-free technical operating specifications 2. Practices aseptic technique	20	14	0.70	0.84

disturb the normal development of surgery. As doctors of different subjects have relatively low satisfaction with cooperation during endoscopic surgery, they established an institution of specialist nurses, which helps these nurses to undertake all endoscopic surgeries and improves the professionalism, proficiency, and satisfaction of cooperation of endoscopic surgeries.

5 Conclusion

Knowledge tree reveals the internal relations of organizational knowledge and states the process of continuous management improvement, as well as the process of corresponding knowledge innovation. Therefore, it provides a kind of management tool for organizational knowledge management, and its value provides a measure for knowledge. According to the use of a knowledge tree, an evaluation system of knowledge management is established.

Using the knowledge tree model of continuous management improvement presented in this chapter, integration of information flows, knowledge flows, and value flows is realized. The model has been applied to hospital service quality control and management innovation and the operational effect of the system is satisfactory.

Acknowledgment Supported by the National Natural Science Foundation of China No. 60874075, and Natural Science Foundation of High Education of Jiangsu Province of China No. 07KJB520139.

References

1. Bao Zhen-Qiang, and Wang Ning-Sheng. A knowledge tree model for managing organizational knowledge. *Science Research Management*, 2002, (1).
2. Chen Yue-Hua. Comment on the current research. *Exploration of Psychology*, 1999, 2.
3. Dai Wan-Wen, Zhao Shu-Ming, Jiang Jian-Wu, Steve F. Foster. Study on the dynamic model of organizational learning processes in the context of knowledge management: A complex system perspective. *China Soft Science*, 2006, (6):1–9.
4. Mika Kivimaki. Communication as a determinant of organizational innovation. *R&D Management*, 2000, 30(1):33–42.
5. OECD web site. <http://www.oecd.org>.
6. Rivak Kfir. A framework, process, and tool for managing technology-based assets. *R&D Management*, 2000, 30(4):297–304.
7. Sampsa Hyysalo. Learning for learning economy and social learning. *Research Policy*, 2009.
8. Shen Chuan-Bin. Study on implicit knowledge visualized. *Science and Technology Management Research*, 2005, (2): 1–4.
9. Wang Huiling, Han Zhuzhu. Path Analysis of implicit knowledge visualized in knowledge management, 2009, (1): 1–3.
10. Yang Zhi-Feng, and Zou Shan-Gang. Knowledge resources, knowledge stocks and knowledge flow: Concepts, characteristics, measures. *Science Research Management*, 2000, (4).
11. Yu Yi-Hong. Knowledge management and organizational innovation. Shanghai: Fudan University Press, 2001.
12. Zhang Weiguo, Wang Shasha, Luo Jun. Discussion on competitive advantage of enterprise activities based on knowledge management. *Scientific Management Research*, 2007, 2(1):1–4.
13. Zhi-Ping Fan, and Bo Feng Evaluating knowledge management capability of organizations: a fuzzy linguistic method. *Expert Systems with Applications*, 2009, 36.

On the Development of a User-Defined Quality Measurement Tool for XML Documents

Eric Pardede and Tejasvi Gaur

Abstract The capability of *eXtensible Markup Language* (XML) for data representation has been widely accepted by research communities and industries. Even though it can be used for efficient data transfer, many industries look for a more promising language on which to rely when it comes to their important data. An ability to provide good XML data quality is necessary to make this data format more reliable and usable. To measure data quality, the current methods are largely driven by structural and technical factors and often assess data quality impartially, not accounting for contextual factors. It is well known that different data share common quality features: *completeness*, *validity*, *accuracy* and *timeliness*. Nevertheless, the measurement of quality features will be unique, based on the data format. The measurement of quality for XML documents cannot be generalised from quality measurement in other data formats. In this chapter, we describe the development of a user-defined quality metric for XML documents. For implementation, we develop a tool that enables users to control XML data quality. We use a case study in health informatics as the proof of concept.

Keywords Data quality · XML document · User-defined quality tool

1 Introduction

XML (eXtensible Markup Language) primarily facilitates the sharing of semi-structured data across different information systems, particularly via the internet, such as passing data from server to client, machine to machine and application to application. XML is an extraction from *SGML (Standard Generalized Markup Language)* with the aim of performing similar web functions as HTML. Compared to HTML, it gives users more choice and freedom to develop their own tags without

E. Pardede (✉)

Department of Computer Science and Computer Engineering, La Trobe University, Melbourne, VIC 3086, Australia

e-mail: e.pardede@latrobe.edu.au

worrying about the web browser's compatibility. In broader terms, XML is simpler, more flexible and more extendable.

In the past decade, the use of XML as a data format has exceeded its use as a markup language. Many domain-specific standards are now structured in the XML format due to its extendable and self-describing nature. It is no surprise that a large volume of XML documents are created and transmitted over the internet everyday.

Some of the data contain important and sensitive content and therefore, the data quality has to be ensured. Data quality describes the relationship between the data and the portrayal of actual phenomena and the degree of excellence of this relationship. In simple terms, if the data is in the correct format for the purpose for which it is required, then it has high data quality.

Much of the existing work [2, 4, 6] has investigated data quality dimensions in various domains and data models. They agree to certain features: *completeness*, *accuracy*, *validity* and *timeliness*. In this work, we discuss how these dimensions are still applicable to measure the quality of XML documents. Based on these dimensions as well, we implement a user-defined quality measurement tool. This tool can be used to assist decision-making for business processes that use XML document as their data format.

1.1 Roadmap

Following the introduction, in [Section 2](#), we briefly discuss the related work. We describe our solution in [Section 3](#) and its implementation in [Section 4](#). A case study is provided in [Section 5](#) and we conclude the chapter in [Section 6](#).

2 Related Work

We found there was a limited amount of research on XML data quality. However, the majority of this literature discussed various aspects of data quality in traditional relational format. The communities have agreed on the most fundamental aspects of data quality, and we argue that these aspects are also applicable to the XML data format. The only difference is the way to measure these aspects due to the different structure of XML data compared to traditional relational data.

Previous work [2, 4, 6] has listed four data quality dimensions:

- *Completeness* (C) is the extent to which data content is present.
- *Accuracy* (A) is the extent to which data is free from errors.
- *Validity* (V) is the extent to which data items conform to their corresponding value domains.
- *Timeliness* (T) is the extent to which data is recent and up to date.

Each dimension is used to measure the *Quality* (Q) of the data. It is the consolidated effect of all the above characteristics.

Table 1 XML data quality dimensions

	Problem	Example
Completeness	Incomplete data due to a missing important value.	<pre><Medical> <Record> <Name></Name> <.....>*****</.....> </Record> </Medical></pre>
Accuracy	Mismatched tags create errors.	<pre><Medical> <Record> <Name>John</Names> <.....>*****</.....> </Record> </Medical></pre>
Validity	The value does not describe the content accurately. For example, no unit measurement, etc.	<pre><Medical> <Record> <Name>John</Name> <.....>*****</.....> <Height>170</Height> </Record> </Medical></pre>
Timeliness	Updated value is not incorporated correctly into the data. For example, the value of the account has not been updated.	<pre><Medical> <Record> <Name>John</Name> <.....>*****</.....> <Height>170 cm</Height> <Account>(-) \$170</Account> </Record> </Medical></pre>

A similar measure can also be applied to the XML data format (see Table 1). For example, a completeness dimension has the same meaning wherever it is used, yet can have different contexts and representation. How we measure complete data in a relational format will be different to the way it is measured in a tree structure format.

Table 2 summarises the existing work in the area of data quality. This work was applied to different domains/applications, used different data format and applied different approaches to determine data quality. Each is unique, but all the solutions are based on the same data quality dimensions. We have summarised all the work, which lists their applications, their respective approaches and the respective data used.

Table 2 Existing works

Applications	Approach	Database/data format
Decision support systems [9]	Visualisation	Relational
Case-based reasoning systems [3]	Goal-question metrics	Heterogeneous
e-Business [5]	Case study on online processing	Relational
Web services [7]	Query based	HTML, XML
Data warehouse [1]	Empirical database	Relational
Health care [8]	Model driven	Relational

Only a small amount of research measures the quality of web services, which were naturally built using XML representation. However, this work cannot be used for a quality measurement tool for XML databases and XML applications.

3 User-Defined Quality Approach

In this section, we propose a solution to measure XML data quality. The solution includes a proposed metric and an algorithm to incorporate the metric into XML documents.

A user-defined metric enables users to determine quality features that a set of XML documents have to follow. In this metric, every element will be given a weight, each of which is variable according to the user's needs. In addition to the weight, a user should be able to provide additional property to check the XML data, for example, an option to provide preference units to an element.

The following metric formula uses all this information to measure the quality factor of XML documents:

$$\text{Quality} = \sum_{r=1}^n \left(\frac{\sum_{vt=1}^k (N_{(vt)} \times \text{weight}_{(vt)})}{\sum_{t=1}^l (N_{(t)} \times \text{weight}_{(t)})} \right)_r \times 100\%$$

where r is the number of records in the document; $N_{(vt)}$ is the number of valid tags in the record; $N_{(t)}$ is the number of tags in the record; $\text{weight}_{(vt)}$ is the user-defined weight for a valid tag; and $\text{weight}_{(t)}$ is the user-defined weight for a tag.

We apply our quality metric in Algorithm 1. This algorithm takes an XML file as an input. At first, it checks the document for all the starting and ending tags. Once all are found, the system concatenates the XML document and stores it in an array in the form of a text file.

The data quality checking procedure starts and the system will check the document against the user-defined metric attribute and their respective units (Line 1-12 to 1-19). For each valid metric attribute, it adds up its respective weight to the total weight of the documents. After the complete document is checked, all the values are entered into the final data quality metric and the document's data quality is calculated.

Algorithm: Counting Quality factor

```

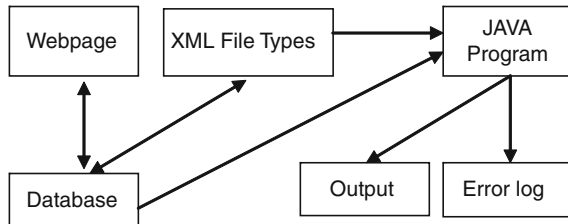
1-1: Input
1-2:  $doc$  = XML Documents
1-3:  $w, p$  = weight and property of the elements  $ele$ 's in  $doc$ 
1-4: Output
1-5:  $Q$  = data quality figures in %
1-6:  $Log$  = error log
1-7: Begin
1-8: Let  $W$  be the total of  $w$ 
1-9: Let  $\mathcal{E}$  is the list of elements  $ele$  in  $doc$ 
1-10: WHILE not end of file of  $\mathcal{E}$ 
1-11:   Check  $ele$ 's tags.
1-12:   IF valid
1-13:     IF  $ele$ 's property doesn't match  $ele$ 's  $p$ 
1-14:       Reduce  $ele$ 's  $w$  from  $W$ 
1-15:       Concatenate the problem description in  $Log$ 
1-16:     END IF
1-17:   ELSE
1-18:     Reduce  $ele$ 's  $w$  from  $W$ 
1-19:     Concatenate the problem description in  $Log$ 
1-20:   END IF
1-21: END WHILE
1-22: Calculate  $Q$ 
1-23: Return  $Q, Log$ 

```

4 Data Quality Checking Tool Implementation

We design a tool for users to define their own quality criteria for their XML documents. The development of the tool follows the diagram in Fig. 1.

Fig. 1 Design model of data quality checking tool



The prototype program is a JAVA-based program which connects to a MySQL database and generates two outputs: (i) *output.txt*, which contains the breakdown of the XML document in a well-structured form after the program reads the values from the tags and (ii) *error.log*, which contains all the details of the XML document that affects its quality. To populate the MySQL database, we use a web interface. The interface will be used to manage the XML documents' properties and the quality factors. The summary of the implementation setup is shown in Table 3.

Our web interface takes the user-defined values for each attribute/element in the XML document (Fig. 2). The user has the freedom to delete and change the entered metric attributes. The attribute measurement units and their weights can be left blank, in which case, default values are used.

Table 3 Implementation setup

Languages used	XML, Java, PHP, HTML
Database used	MySQL database
Input file types	XML files
Output file types	Text files, command line outputs
Drivers used	MySQL-connector-Java-5.1.6-bin.jar
Server used	WAMP server (only for local development and testing purposes)

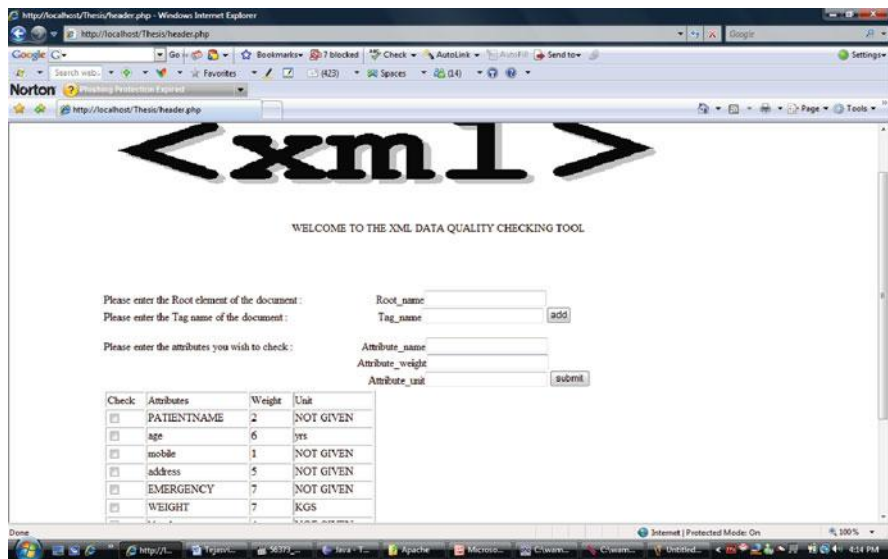


Fig. 2 XML data quality tool web interface

At the current stage of the implementation, the prototype can take reasonably large size of database with textual content. For typical XML data set, it can handle up to 100 MB of data without significant performance problem. It is necessary to realise that this prototype has been developed using small hardware resources (Core 2 Duo 2.0 GHz processor, with 2 GB RAM). For full industrial application with a larger set of data, the more powerful hardware should be applied.

5 Case Study

We apply the developed tool in a real case study using health informatics data. The health informatics sector, like many others, is experiencing a large growth in incoming data, due to the increased number of requirements for which the data is used. The increase in available data has also increased the need to maintain data quality.

Fig. 3 User-defined metric attribute check

Check	Attributes	Weight	Unit
☐	PATIENTNAME	2	NOT GIVEN
☐	age	6	yrs
☐	mobile	1	NOT GIVEN
☐	address	5	NOT GIVEN
☐	EMERGENCY	7	NOT GIVEN
☐	WEIGHT	7	KGS
☐	bloodgroup	4	NOT GIVEN
☐	bloodpressure	6	mmhg
☐	heartconditions	4	NOT GIVEN
☐	reactions	2	NOT GIVEN
☐	allergies	5	NOT GIVEN

In the case of health informatics or medical data, data quality is even more important than efficiency and speed, as the nature of this data is critical and must be precise. For example, the correct storage of records for patients’ blood types is essential and can be a matter of life or death in an emergency situation.

Below is a sample XML document that contains information on a patient. Using our quality tool, a user can identify the metric attributes that have to be checked, as shown in Fig. 3. The user enters all the properties along with their weights. If no unit is given, then default values are used.

```
<MEDICAL>
  <RECORD>
    <PATIENTNAME>Michael</PATIENTNAME>
    <AGE>20yrs</AGE>
    <MOBILE>0433384056</MOBILE>
    <ADDRESS>24 the Fairway, Greensborough - 3334</ADDRESS>
    <EMERGENCY>Steve-0433765673</EMERGENCY>
    <WEIGHT>65KG</WEIGHT>
    <HEIGHT>180cms</HEIGHT>
    <BLOODGROUP>B+</BLOODGROUP>
    <BLOODPRESSURE>120-65mmHg</BLOODPRESSURE>
    <HEARTCONDITIONS>normal</HEARTCONDITIONS>
    <REACTIONS>N/A</REACTIONS>
    <ALLERGIES>DUST ALLERGIC</ALLERGIES>
  </RECORD>
</MEDICAL>
```

In this case, the users want to employ the following properties for patient data: (i) all the lines are complete and no blanks lines are present; (ii) all the fields are complete and no blank fields are present; (iii) all the given units are present with

the respective attributes; (iv) age has ‘yrs’ as the measurement unit; (v) mobile is represented in ten digits; (vi) weight has ‘kgs’ as the measurement unit; (vii) blood pressure has its measuring units; (viii) for attributes which do not have measurement units provided, default values are present; and (ix) all the attributes have given weights.

Figure 4 shows an outcome of a sample XML document quality measurement. In this case, the document only rates 76% against all the quality factors defined by a user. Parts of the XML documents that do not fit the quality factors are logged for future analysis.



Fig. 4 Measurement outcome sample

Based on the outcome, the users will be able to determine whether the XML documents have met the quality criteria and therefore, can be used for further processing or analysis. The users can also define a different set of quality factors depending on the source or the further use of the XML documents.

In health informatics, different hospitals or clinics might have different facilities and different practices for recording their data. If we want to integrate the data, such as for a decision support system, the measurement tool can be used for screening, in the data preparation stage.

While the tool enables user-defined quality metric, it also opens the subjectivity problem. Questions such as who should make the decision on important attributes and their weights should be based on the organisational policy. This chapter aims to provide a tool, which in most cases cannot be run alone without clear procedure on who should use it and how it is applied to assist the business process.

6 Conclusion and Future Work

Due to the increasing volume of XML data used for various applications, database users need a tool to manage the quality of the data. Defining data quality has

been widely researched over many years, and a set of properties such as completeness, accuracy, validity and timeliness have been set as confirmed data quality dimensions.

Unfortunately, judging data quality using these features can be a tedious task and to the best of our knowledge, there is no tool for measuring data quality, especially for XML data. The need for such a tool has become of the utmost importance since XML data, by its nature, will be used for data sharing and integration and therefore, quality has to be maintained and screened carefully.

In this chapter, we apply quality dimensions for the XML data format and implement it as a quality measurement tool. The quality dimensions are not static and we provide user-defined input features to define the quality attributes and their weight. For proof of concept, we provide a case study using health informatics data and test the quality measurement using our quality measurement tool.

For future work, we will incorporate more sophisticated business quality factors into the tool. We will also perform more scalability evaluation of our data quality tool especially if they have to measure the quality of a large batch of XML documents, such as the XML Warehouse. In addition, a user-defined quality formula can be included in our quality tool.

References

1. Ballou, D. P., and Tayi, G. K. (1999) Enhancing data quality in data warehouse environments, *Communications of the ACM* 42(1): 73–78.
2. Batini, C., and Scannapieco, M. (2006) *Data Quality: Concepts, Methodologies and Techniques*. Springer, Berlin.
3. Bierer, A. (2007) Methodological assistance for integrating data quality evaluations into case-based reasoning systems, *Proceedings of the 7th International Conference on Case-Based Reasoning (ICCBR 2007)*, Belfast, Northern Ireland, UK, pp. 254–268.
4. Even, A., and Shankaranarayanan, G. (2007) Utility-driven configuration of data quality in data repositories, *IJIQ* 1(1): 22–40.
5. Paulson, L. D. (2000) Data quality: A rising e-business concern, *IEEE IT Professionals* 2(4): 10–14.
6. Serrano, M. A., Calero, C., and Piattini, M. (2005) Metrics for data warehouse quality, In: Khosrow-Pour, M. (Ed.) *Encyclopedia of Information Science and Technology IV*, Idea Group, Hershey, PA, pp. 1938–1844.
7. Shankaranarayanan, G., and Cai, Y. (2005) A web services application for the data quality management in the B2B networked environment, *Proceedings of the 38th Hawaii International Conference on System Sciences (HICCS 2005)*, Hawaii, USA, pp. 166.
8. Welzer, T., Brumen, V., Golob, I., and Druzovec, M. (2002) Medical diagnostic and data quality, *Proceedings of the IEEE Symposium on Computer-Based Medical Systems (CBMS 2002)*, Maribor, Slovenia, pp. 97–101.
9. Zhu, B., Shankar, G., and Cai, Y. (2007) Integrating data quality data into decision-making process: An information visualization approach, *Proceedings of the 12th International Conference HCI International (HCII 2007) Part I*, Beijing, China, pp. 366–369.

The Paradox of “Structured” Methods for Software Requirements Management: A Case Study of an e-Government Development Project

Kieran Conboy and Michael Lang

Abstract This chapter outlines the alternative perspectives of “rationalism” and “improvisation” within information systems development and describes the major shortcomings of each. It then discusses how these shortcomings manifested themselves within an e-government case study where a “structured” requirements management method was employed. Although this method was very prescriptive and firmly rooted in the “rational” paradigm, it was observed that users often resorted to improvised behaviour, such as privately making decisions on how certain aspects of the method should or should not be implemented.

Keywords e-Government systems development · Requirements management · Requirements prioritisation · Method enactment · Situated action

1 Introduction

Most information systems are “imperfect” for the plain reason that requirements are generally the subject of ongoing negotiation, meaning that at any given time there will be a queue of requirement requests submitted by users who will never quite have everything that they want [19]. This is especially true of the volatile environments that typically characterise web-based information systems (WIS) development, especially e-commerce and e-government projects.

There are a number of methodologies that fall under the general classification of “requirements prioritisation” which aim to produce a best possible wish list under constrained circumstances. In practice, requirements management methods are not executed in the structured fashion typically advocated by their evangelical proponents [12]. This is not surprising, as the underlying philosophy of most of

K. Conboy (✉)

Business Information Systems Group, J.E. Cairnes School of Business & Economics, NUI Galway, Galway, Ireland
Department of Accountancy & Finance, National University of Ireland, Galway, Ireland

e-mail: kieran.conboy@nuigalway.ie

these approaches is that systems development is a rational process, whereas in actuality it is more accurately characterised as “situated action” [25], “improvisation” [6] or “amethodical” behaviour [1]. Herein are reported the findings of a case study of a WIS development project which uses an in-house requirements prioritisation methodology on an e-government system in Ireland. Our objective was to gain insights into the apparently paradoxical situation where developers are seen to engage in rational/methodical and improvised/amethodical behaviour at the same time.

The structure of this chapter is as follows. Section 2 outlines two alternative perspectives of information systems development (ISD): those of “rationalism” and “improvisation” and describes the major shortcomings of each. Section 3 then discusses how these shortcomings were encountered in the case study. Interestingly, although the methodology in this case study was very specific and firmly based in the “rational” paradigm, the users often improvised by privately making decisions on how certain aspects should or should not be implemented. As such, elements of both rationalism and improvisation were experienced, as discussed in the conclusions.

2 Alternative Perspectives of Information Systems Development

2.1 *Technical Rationalism and the “Methodologies Era”*

There is a particular perspective on purposeful human activity, that of “technical rationalism”, which views it as “instrumental problem solving made rigorous by the application of scientific theory and technique” [21] (p. 21). According to the “rational actor” model of decision making, man is an intelligent being whose every action is based on conscious, logical reasoning which he can readily explain. Technical rationalism is founded on the epistemology of positivism, the classical basis of the natural sciences. It was this ideology which famously underpinned Taylor’s *Principles of Scientific Management* [26], and later inspired the 1960s “design methods movement” in the fields of architecture and industrial design, the basic premise of which was that predictability could be introduced into design processes by adhering to rigorous, objective, scientifically founded principles and procedures to solve design problems. Interestingly, many of the chief proponents of systematic design methods in architecture and industrial design later rejected them. Alexander [1] declared that he disassociated himself from the methods movement, remarking that “there is so little in what is called ‘design methods’ that has anything useful to say about how to design buildings . . . I would say forget it, forget the whole thing”. Jones, too, did an about-turn on his thinking, later criticising the design methods movement as “a mania” which placed too much emphasis on technical rationalism [15].

As though oblivious to the experiences in architecture and industrial design, the field of ISD went through its own “methodologies movement” [2] in the 1970s and

1980s, a period which Avison and Fitzgerald [3] call the “methodology era”. In the view of Hirschheim [12], many of the systems development methods then being proposed were firmly grounded in the principles of technical rationalism:

One seemingly common assumption – observable in much of the information systems literature – is that ISD can be thought of as a largely rational and mostly technical process, undertaken with the help of certain well-tried and proven tools and techniques, which is founded on the tenets of classical science.

Most of the well-known systems development methodologies (SDMs) – such as SSADM, Information Engineering and the “Waterfall” SDLC – are firmly based on the principles of rationalism. Empirical studies reveal that, in reality, SDM usage is rather limited, and those who use SDMs mostly tend to judiciously mix and match combinations and parts of methodologies rather than following all the steps required by a particular methodology [8, 11]. A number of major philosophical and pragmatic problems with the use of formalised SDMs in practice have become evident, which we outlined in the following paragraphs.

First, the rational paradigm, being founded on the epistemological assumptions of positivism, assumes that the problem being solved can be well defined and specified before design starts. Based on this assumption, there arises a tendency to separate the “ends” and the “means”; that is, the finished product and the method used to get there [21]. However, the development of software systems is one of the most complex endeavours that man has ever undertaken, as software by definition does not have a clearly visible structure. Because traditional SDMs are based upon the “fixed point theorem” of finite engineering project timeframes – that is, the assumption that “there exists some point in time when everyone involved in the system knows what they want and agrees with everyone else” – they are doomed to inevitably fall short when confronted by the reality of “living” information systems where requirements are in constant flux [19]. In reality, the “ends” are often undefined and likely to change, and any “means” that assume otherwise, such as a methodology which freezes the requirements long in advance of delivery, is likely to disappoint. After Schön, we mean by the *blindness* problem that whereby the “means” are fixed even though the “ends” are not.

Methodologies are tools to get the job done and ought to be invisible to the task. The general trend within ISD has been an increase in the visibility of tools and methodologies. Many users of methodologies became weighed down with the intricate documentation required, instead of focusing on the actual problem at hand, i.e. “confusing the menu with the meal” [8]. We refer to this as the *methodology visibility* problem.

According to the philosophy of rationality, a perfectly rational person is one who always has a good reason for what he does. In practice of course, this is often not the case. What therefore may happen is what Parnas and Clements [18] refer to as “faking the process”, tied to the concept of accountability. In precarious environments such as characterise e-government systems development, it is clear why rational justification is important because a misguided action could lead to time and cost overruns. Developers therefore follow a specified method or at least are

overtly seen to be following a specified method, as a politically astute safety mechanism that affords an excuse against failure [8]. We refer to this as the *accountability* problem.

Robinson [20] highlights another shortcoming of SDMs in practice, that of *theory above practice*. He asserts that the relationship between theory and practice has traditionally been “one of theory standing logically above practice, informing and dictating practice”. Few academics have a sufficiently thorough understanding of the context of real-world systems development. Furthermore, few of the methods they devise are adequately tested in live situations [9]. Nevertheless, academic researchers persist in developing new methods that, not surprisingly given the two previous points, are often impractical and unworkable.

The *definition of rationality* itself is also an issue. Stolterman and Russo [24] distinguish between public and private rationality. If a method is viewed as a way to put specific concepts of rationality into practice, it is obvious that the developer will need to understand the rationale, known as “public rationale”. However, “private rationality” is the term given to the unique judgments and interpretations that a developer will make. These types of rationality may, and probably will, differ.

2.2 Against Rationalism: The Notion of “Situated Improvisation”

Within the literature on decision making and problem solving, the “rational actor” theory has been strongly criticised. Faced by conditions of uncertainty, imperfect information and “ill-structured” [22] or “wicked” [5] problem types, decision makers are more likely to engage in “satisficing” behaviour which aims to produce *acceptable* or *good*, rather than *optimal*, solutions. Simon [23] refers to this as “bounded rationality”, and it is similar to what Lindblom [16] had earlier called “the science of muddling through”. This perspective places greater emphasis on the role of individual judgement, creativity, intuition, and experience in decision making and problem solving.

Striving for predictability and control is a paradox in ISD since the most valued and desirable characteristic of a design process is creativity. In his book *Software Creativity*, Glass [10] strongly sets forth his views that software development is a very complex problem-solving activity that demands a high level of creativity. In reviewing the debate between the need for discipline and formality on the one hand, and the need for creativity on the other, his considered conclusion is that software design, being part art and part science, requires both aspects and one should not suppress the other.

There is a growing body of opinion that ISD resists method and is essentially amethodical [1, 14, 25]. Ciborra [6] refers to an implicit process of improvisation that typifies ISD in practice:

Improvisation is simultaneously rational and unpredictable; planned but emergent; purposeful but opaque; effective but irreflexive; discernible after the fact, but spontaneous in its manifestation.

Externally, design activity may appear as chaotic and maybe even “slightly out of control” [7], but it is guided by the hidden rationality of skilled individuals. Of course, absolute improvisation is a potential licence for anarchy, so improvised approaches should always be founded upon competent decision making, or what Ciborra calls “smart improvisation”. This is situated action that contributes to individual and organisational effectiveness, rather than improvising merely for the sake of dispensing with formalised methods.

Although this theory of “improvisation” is more intuitively satisfactory than that of rationalism, it is not without its difficulties. While formalised methodologies can be overly visible, the opposite is the case here. It is difficult, particularly in “web time”, to record and pass on knowledge gained through iterative processes. The improvisation approach depends on continuity. Large volumes of hard-learned design lessons must be passed from one generation to the next if development processes are to become efficient. Whereas inexperienced developers can learn easily by following rational, prescriptive methods, it is more difficult to pass on knowledge if purely improvised approaches are being relied upon. We refer to this as the *transport of knowledge* problem.

Improvisation can also have a negative impact on project management. By shifting the control towards the developers, it becomes more difficult to tell exactly where a project currently stands. Predicting where an implementation is going and how long it will take to get there becomes harder to determine [16]. This is a critical issue in relation to WIS development, where time is a major constraint. This we refer to as the *shift of control* problem.

A third issue is that whereas rational approaches attempt to eliminate biased judgment or opinion, improvisation can actually encourage developers to “embrace their biases to the point that alternative views are occluded” or to “inflate the importance of their own point of view at the expense of others” [16]. We refer to this as the *bias* problem.

3 Findings of Case Study

This case study is based on a WIS development project conducted by a multinational IT consulting organisation on behalf of a government department. In the interests of confidentiality, the pseudonyms ABC Consulting and GovDept shall be used to refer to the consulting organisation and the government department, respectively.

The purpose of the WIS under development was to facilitate the online registration of births, marriages and deaths. The primary users of the system were registrars located at various bases nationwide, such as hospitals and government offices. However, the general public can also use the system to obtain various certificates and information.

This project was selected for a case study because it exhibited many of the classical characteristics of WIS development environments, as regards the pressures

of accelerated “web time”, an outward extra-organisational focus, a diversity of professional disciplines in the development team and application complexity. The project lasted 6 months, and the development team was quite small, with a maximum of 12 members at one stage during the final build and test. Team members “rolled on” and “rolled off” the project at various times. No specific person performed a requirements management role; rather, different team members elicited requirements particular to their individual area of expertise.

The methodology used was a requirements prioritisation methodology devised in-house by ABC consulting which used a 500-point marking scheme and was administered across various groups and sub-groups. Although this methodology was very specific and firmly based in the “rational” paradigm, the users often resorted to improvisation and privately made decisions on how certain aspects should or should not be implemented. As such, elements of both the methodical (rational) and amethodical (improvised) approaches were experienced. In this section, we discuss how the shortcomings of each approach, as described in Section 2, were experienced.

3.1 Shortcomings of a Methodical (Rational) Approach

3.1.1 Methodology Visibility

In this case, the GovDept was undergoing a decentralisation process, resulting in a distance of 100 miles between the client and the requirements acquisition team. Both the development team and the users found distributed collaboration very difficult. The project manager said that this is always a problem, but the users felt that the rigidity of the methodology did not ease matters and placed an unnecessary burden on them. A particular concern for users was being forced into groups, who had to then facilitate times and locations when every member of a group could attend. For example, there is one registrar in each hospital throughout the country. Therefore, a registrar group meeting forced one person from every hospital in the country to meet at the same location. The users felt that this was rather unnecessary and cumbersome, as one registrar could have adequately prioritised requirements with the same degree of accuracy.

Other users felt that the organisation of users into sub-groups was pointless, given that individual roles within these sub-groups meant that requirements would be prioritised by the most expert user, whose opinion was rarely questionable and met little resistance.

Understandably, the attitude of the users did not change for the better when various inconsistencies with the usage of the methodology came to their attention. Also, they found it hard to understand why such strict guidelines regarding attendance and marking were in place, when finalised requirements were selected in an abstract manner and did not have a well-defined cut-off point as would be expected of any good prioritisation process.

3.1.2 Accountability

Accountability was an important factor in the methodology process. Because the client was a government body, the usual attention to detail and justification was required. This explains the visibility issues earlier outlined. Even though one registrar fulfilled the same role as many others, it was insufficient to rely on the opinion of a single user or even a few similar users. Unwillingness on behalf of the client to depend on the prioritisation process resulted in a refusal to use a strict cut-off point. They were willing to accept the results as a strong indicator of user priorities but not as a basis on which to include or dismiss requirements.

3.1.3 Theory Above Practice

The methodology aimed to produce an ordered ranking of requirements. However, there were major gaps between the prescribed methodology and what was really needed in practice. According to the project manager, these gaps were only apparent in hindsight.

An analysis of the outcomes of the prioritisation process showed that many stakeholders gave the same priority to a number of different requirements. This caused a lot of problems as it became very difficult to discriminate between requirements because of clustering. When each groups’ prioritisation document was examined, over 80% of requirements on each document were not uniquely prioritised. For example, one group awarded 1 mark of 100, 8 marks of 50 – thus fully absorbing the quota of 500 points – so all the remaining 118 requirements were awarded marks of 0.

The project manager felt that the detailed steps relating to the 500-point scale were insufficient. He felt that the users had trouble using “absolute values”, and that every group seemed to award marks in a different way. For example, one group prioritised 97 requirements with an average mark of 5.17. Another group only prioritised nine requirements, giving an average mark of 55.5. This hindered the accuracy and validity of statistical analysis, as a group who rank such a low number of requirements will ensure those requirements are highly ranked in the consolidated list regardless of what other groups try to do. Although the project manager realised the fallacies of the methodology, he could not think of any practical way to improve it.

3.1.4 Definition of Rationality

There was substantial evidence of a conflict between public and private rationality, where the users’ interpretation of the methodology differed from its intended use. Each group of individuals were asked to “rate the following requirements in order of importance” using the 500-point marking system. The conflict occurred in what the individuals deemed as “important”. Some groups interpreted “important” as a requirement that was important to them or their role within the organisation. Others defined “important” as a requirement that was important to the organisation as a

whole, irrespective of its direct impact on them. Legal issues were an example of the latter, where everybody knew how critical legal compliance was, even though the legal team were the only people with direct legal knowledge.

3.2 Shortcomings of an Amethodical (Improvised) Approach

3.2.1 Transport of Knowledge

Users' rationale behind the ratings was not recorded for future reference. When it was found that the users were applying different rationale to their marking systems, and some were using underhand tactics, it was difficult to identify the rationale of each user and to determine exactly what went wrong, because each user's script comprised only of a list of numbers without any explanations.

3.2.2 Bias

A status meeting after the process was completed suggested that the users exhibited an element of bias when prioritising requirements, indicating a flaw in the 500-point rating system. The project manager found that a high number of users gave zero points to certain requirements, because they knew that these requirements would receive adequate marks from other users and groups. This left them with more points to distribute to the requirements that they preferred.

4 Conclusions

4.1 Shortcomings of Methodical Approaches

Because blindness only becomes a problem where the objectives change during a project, which did not happen in this case study because the objective was simply to "get an overview of stakeholders' priorities", it can be said not to apply in this case study. That aside, every other foreseeable shortcoming of the methodical approach was experienced.

It is easy to understand why methodologies developed from theory would succumb to these shortcomings. Many researchers develop approaches without the hindsight of practical implementation and so would be unable to evaluate such factors as the impact of human fallacies. The revelation in this study is that an in-house methodology, created and updated by practical experience alone, not directly influenced by any academic involvement, still fails to overcome these problems. The reason, according to the project manager, is that these problems occur in every project and that they are impossible to overcome.

Two conclusions can be drawn from this study of methodical shortcomings:

- Even an in-house methodology, developed from practice and subjected to repeated validation and revision from numerous projects, did not eliminate methodical shortcomings.
- Managers consider these shortcomings to be a quite natural feature in the implementation of a structured, methodical approach, and efforts to eliminate them are futile. Therefore, no effort was made.

4.2 Shortcomings of Amethodical Approaches

It is remarkable that even though the approach taken in the case study was of a highly methodical nature, two of the three shortcomings of amethodical approaches as earlier outlined were experienced.

There is evidence to suggest that the amethodical shortcomings are encountered for the same reasons methodical ones are. They are simply too difficult to eradicate entirely. When discussing these shortcomings and the reason for their occurrence, the manager again dismissed them as problems that occur as part of every project.

There is evidence to suggest that even though every attempt was made by ABC Consulting to achieve a purely methodical approach, they were trying to achieve an impossible goal. Academic research indicates that most approaches in industry lie somewhere between being purely methodical and purely amethodical in nature. There is a consensus that the advantages and disadvantages of each have driven project managers and method developers to adopt an approach that finds some middle ground between the two. In this case, however, there was no conscious decision to find that middle ground. Instead, every attempt was made to adopt a purely methodical approach to such an extent that the project manager considered the approach taken in this project to be purely methodical and structured in nature. However, there are examples to show that each amethodical shortcoming was encountered simply because a purely methodical approach was not achieved.

In hindsight, it is clear that amethodical tendencies arose where insufficient controls were in place. However, it is very difficult, perhaps impossible, to predict such problems and to then be able to resolve them. The conclusion to be drawn from this is that in a turbulent development environment with so many variables and the unpredictable nature of human actions, it is impossible to develop a completely methodical approach that can control every situation.

References

1. Alexander, C. (1971) The State of the Art in Design Methods. *Design and Manufacturing Group (DMG) Newsletter* 5(3): 3.
2. Avgerou, C. and Cornford, T. (1993) A Review of the Methodologies Movement. *Journal of Information Technology* 8(4): 277–286.

3. Avison, D. E. and Fitzgerald, G. (1999) Information Systems Development. In: Currie, W. L. and Galliers, B. (eds) *Rethinking Management Information Systems*, pp. 250-278 Oxford University Press.
4. Baskerville, R., Travis, J. and Truex, D. (1992) Systems without Method: The Impact of New Technologies on Information Systems Development Projects. In: Kendall, K. E. et al. (eds) *IFIP Transactions A8, The Impact of Computer Supported Technologies on Information Systems Development*, pp. 241-269. Elsevier Science Publishers (North-Holland).
5. Buchanan, R. (1992) Wicked Problems in Design Thinking. *Design Studies* 8(2): 5-21.
6. Ciborra, C. U. (1999) A Theory of Information Systems Based on Improvisation. In: Currie, W. L. and Galliers, B. (eds) *Rethinking Management Information Systems*, pp. 136-155. Oxford University Press.
7. Cusumano, M. A. and Yoffie, D. B. (1998) *Competing on Internet Time/Lessons from Netscape and Its Battle with Microsoft*. New York: The Free Press.
8. Fitzgerald, B. (1997) The Use of Systems Development Methodologies in Practice: A Field Study. *Information Systems Journal* 7(3): 201-212.
9. Fitzgerald, G. (1991) Validating New Information Systems Techniques: A Retrospective Analysis. In: Nissen, H.-E. et al. (eds) *Information Systems Research: Contemporary Approaches and Emergent Traditions*, pp. 657-672. Elsevier Science Publishers B.V. (North-Holland).
10. Glass, R. L. (1995) *Software Creativity*. Englewood Cliffs, NJ: Prentice-Hall.
11. Hardy, C. J., Thompson, J. B. and Edwards, H. M. (1995) The Use, Limitations and Customization of Structured Systems Development Methods in the United Kingdom. *Information and Software Technology* 37(9), 467-477.
12. Hirschheim, R. (1992) Information Systems Epistemology: An Historical Perspective. In: Galliers, R. (ed) *Information Systems Research: Issues, Methods and Practical Guidelines*, pp. 28-60. Oxford: Blackwell Scientific Publications.
13. Hooks, I. and Fellows, L. (1998) A Case for Priority - Classifying Requirements. In: *Proceedings of International Council on Systems Engineering 8th Annual Symposium*, Vancouver, British Columbia, Canada, July 26-30, 1998.
14. Introna, L. D. and Whitley, E. A. (1997) Against Method-ism: Exploring the Limits of Method. *Information Technology & People* 10(1), 31-45.
15. Jones, J. C. (1977) How My Thoughts About Design Methods Have Changed During the Years. *Design Methods and Theories* 11(1): 48-62.
16. Lindblom, C. E. (1959) The Science of 'Muddling through'. *Public Administration Review* 19(2): 79-89.
17. McPhee, K. (1997) *Design theory and software design*. Technical Report TR-96-26. Department of Computing Science, University of Alberta, Edmonton, Canada. May.
18. Parnas, D. L. and Clements, P. C. (1986) A Rational Design Process: How and Why to Fake It. *IEEE Transactions on Software Engineering* 12(2): 251-257.
19. Paul, R. J. (1994) Why Users Cannot 'Get What They Want'. *International Journal of Manufacturing Systems Design* 1(4): 389-394.
20. Robinson, H. (2001) Reflecting on Research and Practice. *IEEE Software* 18(1): 110-112.
21. Schön, D. A. (1984) *The Reflective Practitioner: How Professionals Think in Action*. New York: Basic Books.
22. Simon, H. (1973) The Structure of Ill-Structured Problems. *Artificial Intelligence Review* 4: 181-201.
23. Simon, H. (1981) *The Sciences of the Artificial*, 2nd edition. Cambridge, MA: MIT Press.
24. Stolterman, E. and Russo, N. (1997) The Paradox of Information Systems Methods: Public and Private Rationality. In: *Proceedings of the 5th British Computer Society Conference on Information System Methodologies*, Lancaster, England.
25. Suchman, L. A. (1987) *Plans and Situated Actions*. Cambridge University Press.
26. Taylor, F. W. (1911) *The Principles of Scientific Management*. New York: Harper and Row.

The Research for Knowledge Management System of Virtual Enterprise Based on Multi-agent

Yang Bo and Shenghua Xu

Abstract By analyzing the features and knowledge management system of virtual enterprise, the research introduces the complex adaptive systems into the knowledge management system of virtual enterprise. It offers a model based on the knowledge management system of virtual enterprise and discusses the functions of each agent as well as mutual communication and coordination mechanism.

Keywords Virtual enterprise (VE) · Knowledge management (KM) · Multi-agent

1 Introduction

The concept of virtual enterprise was first put forward from a research report called 21st Century Manufacturing Enterprises Strategy: an industry-led view. Kenneth Preiss, Steven L. Goldman, and Roger N. Nagel were three academicians from Lehigh and Iacocca Institute; they finished the research by cooperation [1]. VE is composed of two or more members, for the time being composed of non-immobilized interdependence, trust, and cooperation between the union and thus the least-cost, the fastest growing market opportunity for quick response. This dynamic alliance relationship is dissolved with the end of product or the project, and members will resume the next round of the dynamic combination of the course. Knowledge management (KM) is a dynamic business environment in order to develop and maintain a competitive edge and knowledge of the production, processing, dissemination, and application of activities such as managing the process. Knowledge is the key to agile VE resources; KM is a VE core competency to achieve a natural choice. However, compared with the traditional enterprise, if VE carries out KM, it not only demands members to equip the independence of KM system in traditional enterprise

Y. Bo (✉)

School of Information Management, Jiangxi University of Finance & Economics, Nanchang, China

e-mail: yb_lh@126.com

but also needs to enhance the function of KM based on VE. VE is a dynamic organization. The members of VE are composed, cooperated, finished, and disbanded for programs and recombined for the next program. With different programs, the partners are different, and it also needs the KM to satisfy the dynamic complexity of VE [2]. The KM system of VE is a dynamic, easy-to-reconstruct, and easy-to-expand system which can solve the distributed heterogeneous problem of sub-system. At present, researchers from home and abroad have offered lots of models of KM system [3–5]. Based on these models and combining the features of VEKM, it can enhance the expansibility, flexibility of VEKM, as well as the functions of personalized and intelligent system. Furthermore, it can also integrate information with original enterprises. This chapter offers a model of VEKM based on multi-agent.

2 The Features of Virtual Enterprise and Knowledge Management

2.1 The Features of Virtual Enterprise

Each member of VE just does their own works for their common interests. Members cooperate with each other and make intellectual property, skills, and information and resources into the paid sharing. Once the product or project has finished, members will be dismissed automatically or restart another dynamic combination [6]. Members of the VE may be scattered in different regions, and they are linked with the technology of Internet and virtual reality and artificial intelligence. All the members have their own abilities and the good sides. VE exists with the market opportunities. VE will compose quickly if the market opportunities appear. If not, then VE will be dismissed until the next cooperation arrives. Anyway, VE has the features of temporality and dynamicity. The members of VE trust each other, and this kind of trust is constructed on some shared interests and goals. The structure of VE changes from structure level to flat and easy-to-reconstruct structure.

2.2 The Features of Virtual Enterprise Knowledge Management

Knowledge process management is the reflection of virtual enterprise knowledge management. The business activities and operations of VE are the results of work cooperation while the knowledge exchange and share among enterprises sustain the whole complex process. In other words, it is a way to create, capture, arrange, transfer, share, and manage innovational by using knowledge in VE. The knowledge and information in the process present a form called knowledge flow.

The KM of VE includes the inner KM of members and knowledge management network between enterprises. The knowledge management of VE tends to knowledge management network while the KM among members is the basis of KMN. KMN is the production of members' cooperation and collective knowledge among

members. It is realized by integrating and activating the knowledge [7]. VE is a dynamic organization. The members of VE are composed, cooperated, finished, and disbanded for programs and recomposed for the next program. With different programs, the partners are different, and it also needs the KM to satisfy the dynamic complexity of VE. The general application of network technology, component technology, artificial intelligence technology, etc. and the culture environment of enterprise support VEKM. By the support of some basic system modules, perfect KM could finish the process to collect, arrange, spread, and apply the knowledge.

3 Multi-agent System and Virtual Enterprise Knowledge System

Agent is considered as a kind of computing entity or program which can apperceive and adapt to the environment as well as to realize some purposes flexibly. According to consumers or other purposes, agent can execute information collection and take apart information into structured data structure, then store it in the repository. In the task under mission control module, it obtains some necessary information to finish the task management and returns the results to consumers or other agents by communication module.

For restrictions on the number of system resources and the ability of agent itself, the single agent can just complete limit missions. The research of multi-agent system does complete the complicated task by a number of self-agents coordinating their efforts. The MAS is a distributed intelligent systems and approaches problems through communication, exchange, cooperation on the Internet, revising its behaviors with the environmental changes. Currently, existing information extraction system can easily embed in MAS and apply wrappers technology to constitute wrappers within it. Consequently, it is possible to require other agent forms to be shifted into existing system ones.

Ontology [8] is a model of the concept of sharing a clear formal specification. Its aim is to find out relevant knowledge of the field, to provide mutual understanding, to establish recognition of common words from different levels, and to give a clear definition of these words (terms) and the inter-relationship between them. In the context, ontology is used to represent knowledge and information. Due to an open and heterogeneous environment, members of the different inter-firm knowledge management system and the original members of the internal systems of knowledge have grammar and semantic differences. To solve the problem of interaction and collaboration between different systems and agent, ontology has to be nested in the agent system, realizing knowledge sharing and reuse. To make agent interaction in the system more efficient and accurate, a system of ontology library is established. Meanwhile, in order to maintain the independence of each agent, ontology is responsible for the use of agent and agent communication between the environment and the exchange of information.

MAS is more adaptable to the environment than agent. KMS is also a distributed system. In order to fulfill the overall objective of virtual enterprises, it is essential

that members cooperate with each other. Under this circumstance, communication and coordination become extremely important. At the same time, MAS has scalability to the changes of its participants and the increase of the number of knowledge. Agent reflects preferences of users with its simple and human way and communicates with other agent users on behalf of certain users. Such trait is particularly suitable to intelligent search and heterogeneous environment of distributed data mining complex as well as knowledge management model.

4 The Virtual Enterprise Knowledge System Model Based on Multi-agent

4.1 The System Model

According to the features of VEKM, the text designs some models of VEKM based on multi-agent. Members of the enterprise knowledge management system model (Fig. 1) and virtual business-to-business networking knowledge management model (Fig. 2) are included in them.

Several kinds of agent are designed in virtual business-to-business networking knowledge management model.

User agent: It can offer users' interfaces which interact with system, according to users' interests and needs to construct users' interests file model as well as produce users' interests theme library. Furthermore, users can make personal customize to knowledge and enhance the precision when users search the knowledge. It can both finish the tasks that users offer passively and follow users' interests and needs automatically. It will initiatively offer appropriate knowledge to users at times and automatically keep and clean part documents in the cooperation work to reduce the burden on users.

Expert agent: It is a special user agent. Except for those functions of user agent, it also equips to some functions of transmitting knowledge. Its major task is to extract

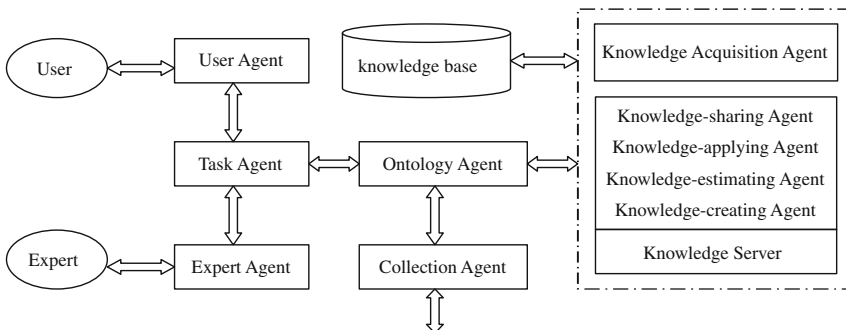


Fig. 1 KM model in members of VE based on multi-agent

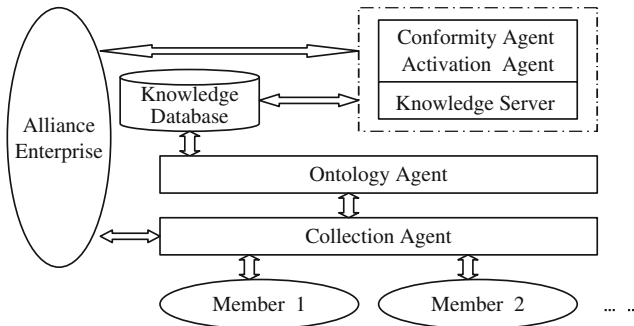


Fig. 2 Networked KM model among VE members based on multi-agent

some hidden knowledge from experts' minds and store them into knowledge base after processing them. At the same time, it can offer a platform to communicate with other agents.

Task agent: This agent receives users' tasks and processes them. It assures some needed knowledge when tasks are finished and offers other agents some services to describe the system. By exchanging with ontology agent and other agents, task agent can acquire necessary knowledge. Keeping watch on the tasks and utilization of knowledge, it functions well. The task is capable of learning automatically and can acquire knowledge in the process to do the task. Furthermore, it can update inner conditions and keep the task knowledge base.

Ontology agent: Ontology agent is a very important part; it can make intelligent visit to one or more information sources (e.g., other database, repository). It has some models which relate to the information sources, resource selection strategies, and conflict resolution mechanism. According to some terms or search of the ontology agent, it can visit many repositories and check some related knowledge.

Collection agent: It collects some relation information, designs searching assignments, and checks the information which has been collected according to users' requests. The agent is a moving agent, which is reactive and proactive. Therefore, it can reflect users' requests, move in every system, and collect information on its own initiative.

Knowledge acquisition agent: It is in charge of grasping knowledge. By using various technologies and rules (e.g., dig data, neural networks, and online analysis), it can dig out knowledge from information sources, then store in certain repository, and inspect and harmonize the actions of task agent. Eventually it realizes the transition and the flow of knowledge through the interaction with other agents.

Knowledge-sharing agent: It is mainly in charge of melting knowledge into every part of VE by dividing, delivering, transferring, and sharing. According to the style and construction of knowledge as well as some sharing rules, dynamic determines the style of knowledge sharing. Therefore, it can adopt different styles of knowledge sharing, like knowledge community, knowledge forum, and knowledge alliance.

Knowledge-applying agent: It is mainly in charge of applying knowledge to every activity and every part of enterprises so as to produce the output of knowledge and acquire the value of knowledge. It is a very important purpose for the knowledge management of VE to apply knowledge.

Knowledge-estimating agent: It is mainly in charge of collecting and analyzing the affection of knowledge application, including the changes of market share, profit, and satisfaction of users. It can deliver this information to knowledge acquisition agent and knowledge-sharing agent in order to guide the cycle and increment of knowledge flows.

Knowledge-creating agent: It is mainly in charge of using advanced technologies and rules to analyze and discord original knowledge as well as to create new knowledges, which can update original repository. At the same time, it can construct the interaction and melting environment so as to provide conditions for workers' discussions. Furthermore, it is better for the creation of hidden knowledge.

In the models of network knowledge management of VE, we design knowledge conformity agent and knowledge activation agent.

Knowledge conformity agent: By knowledge-sharing agent and spreading knowledge to combine different bodies, sources, functions of members. So it can form comprehensive knowledge system of VE and store into the repository of VE. The dominate enterprise of VE will be in charge of the management and protection of repository.

Knowledge activation agent: Through the effects of knowledge conformity agent, knowledge-sharing agent, and knowledge creation agent as well as the comprehensive knowledge to offer a platform for members' study so as to make some brink knowledge or valuable knowledge.

4.2 The Correspondent and Cooperation of System Multi-agent

In order to promise that every agent in MAS can have high efficiency to cooperate with each other, every agent must construct a kind of effective language and rural for interacting. In this system, we use ACL and KQML as the language of every agent and conduct ontology library. Actually, ontology library has defined some conceptions and terms in the system so as to solve the description of various knowledge, realize the conception of ontology agent with every agent, and improve the efficiency of knowledge sharing and application.

The cooperation of agents is accomplished by finishing tasks. If users want to check some necessary knowledge, they can do it by the alternation of corresponding agent. The user offers a task to task agent by user agent and task agent will analyze the task to ensure the knowledge which needed to finish the task. After analyzing the task, task agent will correspond with ontology agent and ask for the knowledge server in the visiting system to acquire some necessary knowledge. Ontology agent has alternation with knowledge acquisition agent to acquire some necessary knowledge and transform them. Eventually, knowledge will be returned to the task agent

to carry out the task. The task agent will check and control the process of the task while the result will be offered to the user by task agent according to a certain format. The knowledge server mainly offers all kinds of application operations related to knowledge, like knowledge sharing, knowledge estimating, knowledge conformity, and knowledge activation. If users cannot acquire their required knowledge in the local repository, they can solve the problem by three ways. First, according to the knowledge map in ontology agent, users can orient knowledge in other repositories among members by collection agent to check. Second, they can also orient to some experts in this field and establish the connection between user agent and expert agent so that they can directly interact with experts as well as realize the sharing of hidden knowledge. Third, some users have common interests and purposes, so they can interact and cooperate with each other in order to promote the share of hidden knowledge at maximum.

The integration of original system and knowledge management system is realized by the package technology. Actually, collection agent is a kind of wrapper agent. It is introduced to MAS system by packaging outer software system while other agents deliver visiting order to wrapper agent by ACL. Wrapper agent will analyze the received information and transfer them into the orders of outer software system, then execute them. After that, the results will be returned to the requested agent. The wrapper agent offers other agent a unitize method to visit outer software system and it also gives an effective method to the original system and dynamic integration of KMS.

The key enterprise (alliance enterprise) is in charge of protecting and updating repository and it establishes the sharing knowledge map of VE in alliance ontology agent. By the maps in local collection agent and ontology agent and with the help of the communication between knowledge server of alliance enterprise and repository, members will conform, recombine, and activate the sharing knowledge on the knowledge server of alliance enterprise in order to finish the purpose of whole VEKM.

5 Conclusions

According to the features of virtual enterprise knowledge management system, this chapter introduces distributed artificial intelligence agent technology and puts forward a kind of model based on the virtual enterprise knowledge management system of multi-agent. In this model, members can independently establish their own knowledge management sub-system. And at the same time, members can realize the knowledge recycling flow and increment of the whole VE by the cooperation of ontology and each agent. This model has the features of intensive openness, adaptability, flexibility, and expansibility. It can manage the knowledge of VE effectively. In the future, Java will be used as an exploration language while JATLITE as the exploration saddlebag of agent in order to realize the antitype of this model. Furthermore, it will consummate the detailed design for the service function of ontology as well as the communication interface of agent.

References

1. Preiss, K., Goldman, S. L., and Nagel, R. N. 21st Century Manufacturing Enterprises Strategy: An Industry-Led View. Lacocca Institute, Lehigh University, 1991.
2. Qi, E.-S., Liu, C.-M., Wang, L., and Huo, Y.-F. Research on the Knowledge Management System in Virtual Enterprise. *Document; Information & Knowledge*, 2004, (1): 22–24.
3. Hoog R. Use Your Intranet for Effective Knowledge Management. *e-Business Advisor*, April 1999.
4. Xue, C.-F. and Zhang, J.-S. A Framework of Knowledge Management System in Virtual Enterprise Based on Grid. *Journal of Information*, 2006, (4): 57–60.
5. Xia, H.-S. and Cai, S.-Q. A Framework of Knowledge Management Systems Based on Internet. *China Soft Science*, 2002.
6. Shen, J. and Qing-Song. Building a Model – Model of the Information Resources Management in Virtual Enterprise. Tianjin University Press, 2007.
7. Hu, J.-F. and Du, W. Study on the Knowledge Management of the Virtual Enterprise. *Sci-Tech Information Development & Economy*, 2006, (8): 181–182.
8. Chandrasekaran, B., Josephson, J. R. and Richard Benjamins, V. What Are Ontologies, and Why Do We Need Them?. *IEEE Intelligent Systems*, 1999, 14(1):20–26.

Part IV
Model-Driven Engineering in ISD

Problem-Solving Methods in Agent-Oriented Software Engineering

Paul Bogg, Ghassan Beydoun, and Graham Low

Abstract Problem-solving methods (PSM) are abstract structures that describe specific reasoning processes employed to solve a set of similar problems. We envisage that off-the-shelf PSMs can assist in the development of agent-oriented solutions, not only as reusable and extensible components that software engineers employ for designing agent architecture solutions, but just as importantly as a set of runtime capabilities that agents themselves dynamically employ in order to solve problems. This chapter describes PSMs for agent-oriented software engineering (AOSE) that address interaction-dependent problem-solving such as negotiation or cooperation. An extension to an AOSE methodology MOBMAS is proposed whereby PSMs are integrated in the software development phases of *MAS Organization Design*, *Internal Design*, and *Interaction Design*. In this way, knowledge engineering drives the development of agent-oriented systems.

Keywords Agent-oriented software engineering · Problem-solving methods

1 Introduction

The demand for agent-oriented software has motivated the creation of new development approaches, such as Gaia [1], INGENIAS [2], Tropos [3] and MOBMAS [4]. A number of proposals have been made that consolidate various aspects of development approaches in an attempt to move towards a general agent-oriented approach applicable to most situations [5–7]. None adequately addresses support for the issues of extensibility, interoperability and reuse other than [8] where it has been argued that an ontology-based approach is needed for a truly domain-independent agent-oriented development. Towards this, Beydoun et al. [8] proposed development of agents with *domain-dependent ontologies* designed with *problem-solving methods*

P. Bogg (✉)
University of New South Wales, Sydney, NSW, Australia
e-mail: paulb@unsw.edu.au

(PSMs). PSMs are high-level structures that describe a reasoning process employed to solve general problems [9]. PSMs have been investigated for use in modelling traditional knowledge-based systems [10]. A library of modular, reusable PSMs would assist the domain-independent development of agent-oriented systems, reducing development costs and speeding up the development process.

The work described in this chapter continues on from [8] and extends it in the following ways: First, the design of a PSM library specifically oriented towards agent-oriented software engineering (AOSE) is described with two new constructs: *interaction dependencies* and *interaction-specific PSMs*. *Interaction dependencies* are extensions to existing PSMs within a library, guiding developers towards the design of solutions where interaction is necessary for problem-solving (such as in cooperation and negotiation). *Interaction-specific PSMs* are new additions to PSM libraries, describing interaction-specific problem-solving knowledge that designers may use to design the performance of interactions between agents. Second, these agent-specific PSM libraries are grounded for use in AOSE with a proposed extension to an existing AOSE methodology, MOBMA [4], for the development of multi-agent systems (MAS). The MOBMA extension demonstrates how PSMs might be incorporated in an agent-oriented methodology.

2 Background and Related Work

A general ontology-based structure for PSMs was proposed with UPML [11]. UPML encapsulated previous approaches to describing general task and problem-solving knowledge. One limitation in UPML is the lack of consideration given to PSMs for tasks where multiple software components are required to interact in order to solve a problem. However, interaction-dependent problem-solving (such as negotiation) is prevalent in agent-oriented systems. For instance, in competitive e-commerce contexts, problem-solving agents may be required to negotiate for executing online trades.

In order to leverage the complete benefits of PSMs in AOSE, PSM structures addressing interaction-dependent problem-solving need to be developed. Recent approaches to incorporating PSMs into agent-oriented architectures have not addressed this limitation. MAS-CommonKADS [12] advocates task and problem-solving knowledge representation and PSM use in its methodology; however, it mistakenly presumes the existence of PSM libraries suitable for complex, interaction-dependent problem-solving. The ORCAS framework [13] introduces methods to adapt PSMs to agent capabilities in order to maximize reusability. The ORCAS framework addresses cooperation as “agent teams” at the knowledge level; however, it does not address interaction-dependent problem-solving knowledge such as in negotiation contexts. In some cases, problems to which agents are suited do not have adequately formulated solutions, and PSMs may be used as the basis for re-engineering new methods with domain-specific heuristics [14]. These heuristics however do not address interaction-dependent problem contexts.

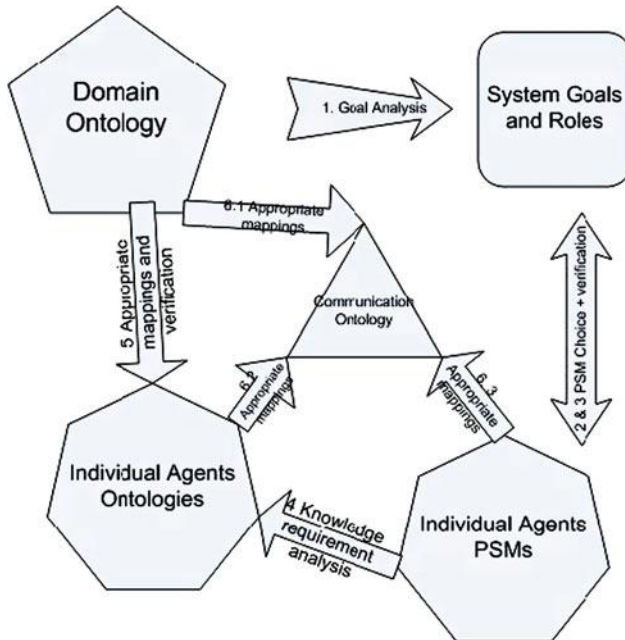


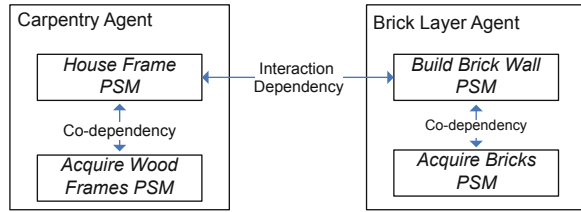
Fig. 1 Ontology-based MAS development using PSMs: (1, 2) Domain ontology produces Goal Analysis which can be used to select PSMs from a PSM bank. (3, 4) Knowledge analysis can then be used to delineate local agent knowledge. (5, 6) Verification against a domain ontology and the formulation of the communication ontology (language)

In addition, none of the above-mentioned approaches have really incorporated PSMs in the detailed design phases of an AOSE methodology. In [15], software engineering requirements to use PSMs were mapped out resulting in a methodological model (Fig. 1) underpinned by the presence of PSM libraries appropriately represented. The work provided a broad conceptual framework to integrate the use of PSMs within an AOSE methodology without actually addressing the issue of how to best formulate the PSMs for interaction-dependent problem-solving. This chapter continues this work by formulating an appropriate way to extend PSM libraries with agent-specific constructs. Furthermore, an extension to MOBMAS uses these agent-specific PSM libraries in AOSE software development.

3 Extending PSM Libraries for AOSE

PSM libraries have been used for the development of single-agent knowledge-based systems – examples include scheduling [16] and planning [17]. These libraries are oriented towards individual problem-solving. Where complex distributed problem-solving depends on extensive interactions between agents, we propose that existing PSM formalism is not sufficient. We propose a new interaction-dependent

Fig. 2 Agent-level PSM composition



problem-solving where further consideration is given to how PSMs within existing libraries relate to one another. Towards this, existing PSM libraries are extended with two constructs: *interaction dependencies* between PSMs and *interaction-specific PSMs* used to design complex interactions for agent-oriented systems.

We also use a modelling construct, a *PSM co-dependency*, to describe a relationship between PSM definitions wherever execution of one PSM's procedure depends on the execution of another PSM's procedure. For instance, in designing agents to autonomously coordinate building a house (Fig. 2), a carpentry agent's PSM for building a house frame may depend on the PSM for acquiring wood. A *PSM interaction dependency* is where dependencies exist between PSM definitions for separate agents, suggesting that interaction between agents is required to execute PSM procedures to solve problems. For instance, in building a house, a brick layer agent's PSM for building a wall is interaction-dependent on a carpentry agent's PSM for building a house frame.

The new formalism of interaction-dependent PSMs may suggest the presence of additional methods and/or agents not previously identified and also assist in designing the interaction structure between agents by suggesting what type of exchange is required between agents. The type of exchange required might be as simple as an enquiry (such as what might exist between the carpentry and brick layer agents in Fig. 2), or as sophisticated as negotiation (such as what might exist between e-commerce trading agents). Since interaction-dependent PSMs are ontology-based, reuse (as suggested in [18]) is a natural feature supporting domain-independent AOSE development.

When interaction-dependent PSMs suggest the exchange between two agents is sophisticated (such as negotiation, coordination or cooperation), *interaction-specific PSMs* may be required. Interaction-specific PSMs are methods specifically oriented towards reusing existing knowledge about how to resolve complex interaction-dependent problem-solving. This development recognizes that knowledge about interaction-dependent problem-solving is reusable in different domains and for different tasks (for instance, a similar method for negotiation in e-commerce trade might be adopted in the negotiation of free trade agreements). We use literature on designing agents for negotiation, cooperation and coordination to identify three types of interaction-specific PSMs (Fig. 3):

- *Interaction Protocol PSM* defines the rules for interaction engagement. It differs from a communication protocol, which defines terms that agents can utter. An

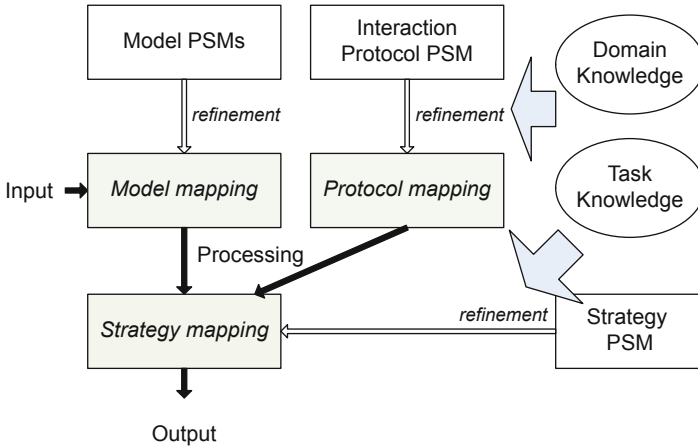


Fig. 3 Knowledge-level interaction-specific PSMs

interaction protocol defines an order to engagements between agents using terms expressed by the communication protocol.

- *Model PSMs* are structured knowledge about how to model information that an agent observes. They directly relate to interaction because agency requires autonomous assessment of itself, external agents and the environment. An interaction protocol may constrain the types of information an agent may observe about agents with which it is interacting. For example, a simple protocol for a negotiation scenario may limit information available with which to model other agents.
- *Strategy PSMs* are structured knowledge about how interactive behaviour is derived from the output of Model PSMs and Interaction Protocol PSM.

The above-mentioned three types of interaction-specific PSMs are derived from classifications of agent design components used for interaction-dependent problem-solving (such as described in [19–21]). For example, [19] describes variations of interaction protocols where particular strategies depend on models of utility for cooperative distributed problem-solving. Lomuscio et al. [21] describes interaction protocols and strategies as the two basic types of components for agent-based negotiation. Jennings et al. [20] describes areas of negotiation research concerned with protocols, negotiation objects, and decision-making models.

4 Extending MOBMAS with PSMs

To illustrate our formalism in a software engineering context, we add explicit support for PSMs in one particular methodology, MOBMAS. MOBMAS [4] is an

agent-oriented methodology that integrates ontologies in the analysis and design of multi-agent systems (MAS). MOBMAS itself comprises five phases:

- *Analysis* – identification of tasks and roles to be performed by the MAS, and ontologies required for the domain.
- *MAS Organization Design* – specification of MAS structure and agent class composition.
- *Agent Internal Design* – specification of each individual agent class, including beliefs, goals, events and plans.
- *Agent Interaction Design* – specification of interactions between agents.
- *Architecture Design* – instantiation of agent classes and the deployment configuration of agents.

We extend MOBMAS at the *MAS Organization Design*, *Agent Internal Design*, and *Agent Interaction Design* phases. In the *MAS Organization Design* phase, PSMs assist both the formulation of agent classes and identification of knowledge types for extending ontology development (post analysis). In the *Agent Internal Design* phase, PSMs are refined as methods for tasks used to achieve agent goals. In the *Agent Interaction Design* phase, interaction dependencies and interaction-specific PSMs are used to design problem-solving interactions. At each phase of development, PSMs may suggest the refinement of the outcomes of a previous phase. For instance, PSMs refined in the *Agent Internal Design* phase may suggest new formulations of agent classes in the *MAS Organization Design* phase.

In the following sub-sections, we assume the existence of two types of PSM-based libraries: a PSM library and a runtime capability library. The former is a collection of PSM definitions (such as those proposed by [16]) that include interaction-dependent PSM definitions and the interaction-specific types of PSMs (protocols, models and strategies) as defined in the previous section. The runtime capability library is a collection of PSM implementations which are code-based capability modules that are modelled directly from PSM models for particular domains and tasks (such as those described in [13]).

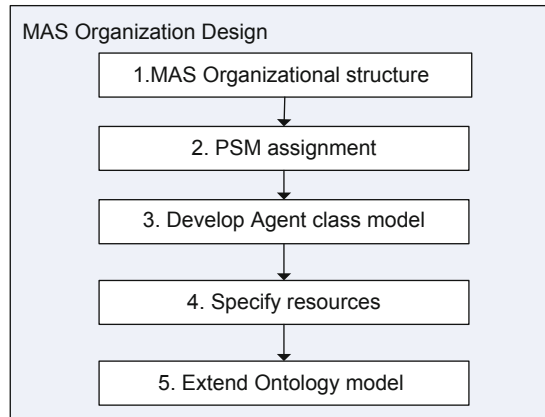
4.1 MAS Organization Design

The MAS Organization Design phase consists of four steps¹: *MAS organizational structure*, *Develop agent class model*, *Specify resources*, and *Extend ontology model (optional)*. We introduce a new step: *PSM assignment*. The *PSM assignment* step assists in the development of the agent class model and may be used to extend the ontology model (Fig. 4).

PSM assignment is performed after the *MAS organizational structure* step. PSM assignment is a preliminary assignment of PSM definitions (selected from a PSM

¹Note that we use the term ‘step’ for a task in a methodology to avoid confusion with agent and role tasks.

Fig. 4 MAS Organizational Design view



library) to tasks identified from roles within the organizational structure. It is necessary to perform this assignment during *MAS Organizational Design* as it assists the development of the agent class model – the consideration of methods to solve a group of tasks may assist in identifying the agent classes. A suggested technique for *PSM assignment* is

1. For each task, search the PSM library to find a suitable PSM definition and *assign* it to the task. If no suitable PSM definition can be found, a new method may need to be designed at the *Agent Internal Design* phase.
2. For each PSM definition assigned to a task, assess existing dependencies with other PSM definitions. A dependent PSM definition may suggest the existence of tasks that were not identified.
3. Iteratively revise the assignment of PSM definitions to tasks (if necessary) to ensure consistency between PSM dependencies.

The work-product of the PSM assignment step is a list of one-to-one assignments from tasks to PSM definitions. This list is a *preliminary* assignment because no effort has been made (yet) to refine PSM definitions towards each specific task for each agent – this is performed in the *Agent internal design* phase. As a consequence of the assignment of PSMs to tasks, missing tasks (or possibly missing requirements) may be identified. This may suggest a revision of the task model in the *Analysis* phase.

The *Develop agent class model* step follows the *PSM assignment* step. Agent class models are built upon roles, whereby one agent may be assigned multiple roles, or multiple agents may be assigned a common role, depending on the requirements. In assisting formulation of the agent class model, PSM assignments may be used as a guideline with two rules of thumb:

- (a) Where tasks have associated PSM *co-dependencies*, they should be grouped together within an agent class.

- (b) Where tasks have associated PSM *interaction dependencies*, they should be in separate agent classes associated by some relationship.

The *Extending ontology model* step is optional depending on whether the identified PSM definitions require additional domain knowledge. *Domain* and *task ontologies* resulting from the *Analysis* phase may be extended by further analysis on knowledge required for each PSM definition. This idea is supported by [22], where it is suggested that PSMs may decrease the knowledge acquisition effort by software engineers by suggesting possible knowledge types to be considered.

4.2 Agent Internal Design

Agent Internal Design phase consists of four steps: *Belief conceptualization*, *Specify agent goals*, *Specify events*, and *Agent behaviour model*. We introduce a new step: *PSM orchestration model* (Fig. 5). The *PSM orchestration model* step confirms the selection of appropriate PSMs and then refines each PSM towards the specific task and domain. The PSM orchestration model outcome assists in the development of the Agent behaviour model.

The *Specify agent goals* step has been defined in MOBMAS to translate tasks into agent goals. The translation of tasks into goals may influence the selection of PSM definitions assigned to each task. The developer is recommended to check that PSM definitions are consistent with the formulation of agent goals.

The *PSM orchestration model* is performed after the *Specify agent goals* step. For each agent, PSM definitions are refined to each agent goal for the agent's domain ontology. A suggested technique is:

1. For each task/goal identified for each agent, select the assigned PSM definition. If there is no PSM definition assigned, a new method may need to be designed – see 3 below.

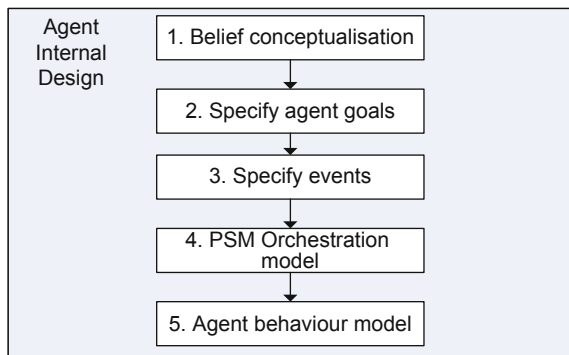


Fig. 5 Agent Internal Design view

2. Using the agent’s domain and task ontologies, *refine* the PSM definition from 1 (above) to the specific task. The result should be a *PSM mapping* tailored to solve the specific task required. The task/goal is now assigned the PSM mapping.
3. If a new method needs to be constructed, evaluate whether this method should be designed as a new PSM (incorporated into a library, and refined using 1 and 2 above), or be a custom method designed for the specific task/goal.

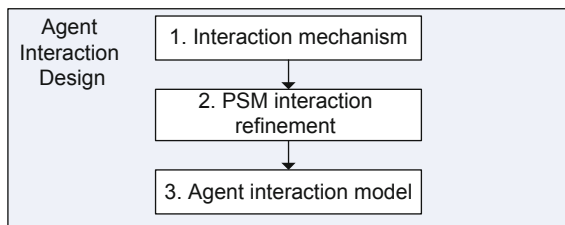
The work-products from this step is a set of *PSM mappings* for the agent that adequately cover every task/goal required by the agent to be performed. The PSM mappings may suggest that the agent class model needs revision. In this case, the PSM mappings may suggest the iterative refinement of the agent class model in the *MAS Organizational Design* phase.

The *Agent behaviour model* step is the last to be performed. PSM mappings are highly domain-specific, task-oriented knowledge structures, and plans are not necessary where task/goals are assigned a PSM mapping. However, for task/goals without assigned PSM mappings, custom methods need to be designed. The design follows the MOBMAS prescription for determining whether plan templates or reflexive rules are selected.

4.3 Agent Interaction Design

The Agent Interaction Design phase consists of two steps: *Interaction mechanism* and *Agent interaction model*. This phase is concerned with the specification of interactions between agents. We introduce a new step: *PSM interaction refinement* (Fig. 6). Interaction dependencies between PSM mappings are used for the development of the *Agent interaction model*. *PSM interaction refinement* should only be performed once developers are certain of the composition of each agent class.

Fig. 6 Agent Interaction Design view



The *PSM interaction refinement* is performed after the selection of the interaction mechanism. The interaction mechanism specifies what communication protocol is to be used. Interaction refinement uses the communication protocol and domain ontologies to identify and refine Interaction Protocols, Model, and Strategy PSMs. A suggested technique is to perform the following:

1. For each PSM mapping, check for interaction dependencies with other PSM mappings; if a simple message exchange is a sufficient interaction, then no

further refinement is necessary; if a simple message exchange is not sufficient, continue to 2 below.

2. For each interaction dependency, for each agent, determine whether any *Model PSMs* are required– co-assigned tasks may assist here.
3. For each Model PSM identified, *extend* the domain ontologies (if necessary) by using the Model PSM to suggest knowledge types not already included.
4. Once all Model PSMs are identified for an interaction dependency, select an appropriate Interaction Protocol PSM that supports the interaction and agent goals.
5. Using the agent's domain and task ontologies, refine the Model PSMs to produce *model mappings* – domain-specific models of information necessary for interaction-dependent problem-solving.
6. Using the communication protocol and agents' domain ontologies, refine the Interaction Protocol PSM to produce a *protocol mapping* – domain- and task-specific interaction rules adhering to a communication protocol.
7. An agent-level consistency check is to be made against the PSM mappings from the *Agent Internal Design* phase. For each agent, the input/output of the protocol and model mappings needs to be consistent with the input/output of the internal design mappings. If there is an inconsistency (such as one output not corresponding to another's input), then revisions may need to be made.
8. Select Strategy PSMs for each agent on the basis of the knowledge types used by the agent's *model mappings*, consistent with the *protocol mappings*, where the strategy adheres to *goals* identified in the *Agent Interaction Design* phase.
9. Refine each Strategy PSM by the domain ontology to the specific knowledge types used by the model mappings, and ensure the output of the strategy is in the language specified by the communication protocol. The work-product is a set of *strategy mappings* for each agent.

The work-products of this step are a set of *protocol mappings*, *model mappings* and *strategy mappings* consistent with the work-products from the *Agent Internal Design* phase.

The *Agent interaction model* step is performed next. Protocol mappings provide structure to agent-to-agent message passing in the language of the communication protocol. This includes what type of parameters can be passed, and the permissible sequences of message exchanges. A strategy mapping for a particular interaction protocol defines a selection of possible messages that an agent might send at a particular point in time. In other words, strategy models provide options for message exchange, and interaction protocol provides message exchange restrictions. The interaction model is defined in terms of the protocol mappings and strategy mappings for each agent.

At the conclusion of the design phase, the formulation of the agent-oriented system should be complete. MOBMAS also advocates an *Architecture Design* phase whereby agent classes are instantiated. In terms of a PSM-based AOSE methodology, a runtime capability library may exist whereby PSM mappings may be used to directly infer runtime capabilities available within the library (a related example

is in [13]). The identification and selection of runtime capabilities for agent class instantiation is not discussed here, and left as future work.

5 Conclusions

The use of ontologies as central artefacts in AOSE has only relatively recently been investigated [12]. To effectively reuse ontologies in AOSE across various problem domains, the problem-solving capacity of agents has to be encapsulated in a reusable way as well. We have described an approach towards addressing the gap between these two complementary aspects of reuse. We introduced interaction dependencies and interaction-specific problem-solving methods (PSMs) as necessary components of PSM libraries for AOSE. Interaction dependencies are used by software engineers in the analysis of solutions to complex problems where interaction is required. Interaction-specific PSMs are used to design agent-oriented systems capable of complex interaction-dependent problem-solving. PSMs are beneficial to AOSE in three ways. First, PSMs encourage reuse of complex problem-solving knowledge during design phases of an AOSE methodology. Second, PSMs support knowledge acquisition by suggesting possibly knowledge types during analysis. Third, PSMs support design of agent class composition. To demonstrate this, MOBMA was extended to incorporate PSMs at the *MAS Organization Design*, *Agent Internal Design*, and *Agent Interaction Design* phases. By incorporating PSMs, knowledge engineering drives AOSE development with domain, task, and problem-solving knowledge.

At present, more work is being performed in producing a more formal description of PSM libraries for AOSE. Reusable interaction-specific PSMs are being designed for problem-solving in negotiation encounters. It is envisaged that negotiation-specific PSM library will assist AOSE development of multi-agent systems for complex problem-solving domains.

References

1. Wooldridge M, Jennings NR and Kinny D (2000) The Gaia Methodology for Agent-Oriented Analysis and Design. In *Autonomous Agents and Multi-Agent Systems* (Jeffrey S. Rosenschein and Peter Stone, eds), pp. 285–312. Kluwer Academic Publishers, The Netherlands.
2. Pavon J, Gomez-Sanz J and Fuentest R (2005) The INGENIAS Methodology and Tools. In *Agent-Oriented Methodologies* (Henderson-Sellers B and Giorgini P, eds), pp. 236–276. IDEA Group Publishing.
3. Bresciani P, Giorgini P, Giunchiglia F, Mylopoulos J and Perini A (2004) TROPOS: An Agent Oriented Software Development Methodology. *Journal of Autonomous Agents and Multi-Agent Systems* **8**, 203–236. Hershey PA, USA.
4. Tran Q-NN and Low G (2008) MOBMA: A Methodology for Ontology-Based Multi-Agent Systems Development. *Information and Software Technology* **50**, 697–722.
5. Fitzgerald B, Russo N and O’Kane T (2003) Software Development Method Tailoring at Motorola. *Communications of the ACM* **46**, 65–70.
6. Glass RL (2004) Matching Methodology to Problem Domain. *Communications of the ACM* **47**, 19–21.

7. Glass RL (2000) Process Diversity and a Computing Old Wives'/Husbands' Tale. *IEEE Software* **17**, 127–128.
8. Beydoun G, Krishna AK, Ghose A and Low GC (2007) Towards Ontology-Based MAS Methodologies: Ontology Based Early Requirements. *Information Systems Development Conference* (Barry C, Lang M, Wojtkowski W, Wojtkowski G, Wrycza S and Zupancic J, eds), pp. 1–13. Galway.
9. Rodríguez A, Palma J and Quintana F (2003) Experiences in Reusing Problem Solving Methods – An Application in Constraint Programming. *Proceedings of Knowledge-Based Intelligent Information and Engineering Systems, 7th International Conference, KES*, 1299–1306. Oxford, UK.
10. Studer R, Benjamins VR and Fensel D (1998) Knowledge Engineering: Principles and Methods. *Data and Knowledge Engineering* **25**, 161–197.
11. Fensel D, Motta E, Benjamins VR, Crubezy M, Decker S, Gaspari M, Groenboom R, Grosso W, Harmelen Fv, Musen M, Plaza E, Schreiber G, Studer R and Wielinga B (2002) The Unified Problem-Solving Method Development Language UPML. *Knowledge and Information Systems* **5**, 83–131.
12. Iglesias CA and Garijo M (2005) The Agent-Oriented Methodology MAS-CommonKADS. In *Agent-Oriented Methodologies* (Henderson-Sellers B and Giorgini P, eds), pp. 46–78. IDEA Group Publishing, Hershey PA, USA.
13. Gómez M and Plaza E (2007) The ORCAS e-Institution: A Platform to Develop Open, Reusable and Configurable Multi-Agent Systems. *International Journal of Intelligent Control and Systems* **12**, 130–141.
14. Angele J and Studer R (2008) A State Space Analysis of Propose-and-Revise. *International Journal of Intelligent Systems* **14**(2), 165–194, February 1999.
15. Beydoun G, Tran N, Low G and Henderson-Sellers B (2006) Foundations of Ontology-Based Methodologies for Multi-agent Systems, vol 3529. pp. 111–123, Springer, Berlin.
16. Rajpathak DG, Motta E, Zdrahal Z and Roy R (2006) A Generic Library of Problem Solving Methods for Scheduling Applications. *IEEE Transactions on Knowledge and Data Engineering* **18**, 815–828.
17. Valente A, Benjamins VR and De Barros LN (1998) A Library of System-Derived Problem-Solving Methods for Planning. *International Journal of Human-Computer Studies* **48**, 417–447.
18. Breuker J (1999) Indexing Problem Solving Methods for Reuse. *Knowledge Acquisition, Modeling and Management* **1621**, 315–322.
19. Sandholm T (1999) Distributed Rational Decision Making. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, 201–258. The MIT Press.
20. Jennings N, Faratin P, Lomuscio A, Parsons S, Sierra C and Wooldridge M (2001) Automated Negotiation: Prospects, Methods and Challenges. *International Journal of Group Decision and Negotiation* **10**, 199–215.
21. Lomuscio A, Wooldridge M and Jennings N (2001) A Classification Scheme for Negotiation in Electronic Commerce. *Agent Mediated Electronic Commerce, The European AgentLink Perspective* **1991**, 19–33.
22. Molina M, Hernandez J and Cuena J (1998) A Structure of Problem-Solving Methods for Real-Time Decision Support in Traffic Control. *International Journal of Human and Computer Studies* **49**(4), 577–600.

MORPHEUS: A Supporting Tool for MDD

Elena Navarro, Abel Gómez, Patricio Letelier, and Isidro Ramos

Abstract Model-driven development (MDD) approach is gaining more and more attention both from practitioners and academics because of its positive influences in terms of reliability and productivity in the software development process. ATRIUM is one of the current proposals following the MDD principles as the development is driven by models and a tool, MORPHEUS, supports both its activities and models. This tool provides facilities for modelling, metamodeling, and analysis and integrates an engine to execute transformations. In this work, this tool is presented describing both its architecture and its capabilities.

Keywords Model-driven development · Requirements engineering · Software architecture

1 Introduction

Software development process is always a challenging activity, especially because systems are becoming more and more complex. In this context, the model-driven development [24] (MDD) approach is gaining more and more attention from practitioners and academics. This approach promotes the exploitation of models at different abstraction levels, guiding the development process by means of transformations, so that traceability and automatic support becomes a reality. MDD has demonstrated positive influences for reliability and productivity of the software development process due to several reasons [24]: it allows one to focus on the ideas and not on the supporting technology; it facilitates not only the analysts to get an improved comprehension of the problem to be solved but also the stakeholders to obtain a better cooperation during the software development, etc. With those aims, MDD exploits models to both properly document the system and automatically or

E. Navarro (✉)

Department of Computing Systems, University of Castilla-La Mancha, Castile-La Mancha, Spain
e-mail: elena.navarro@uclm.es

semi-automatically generate the final system. This is why the software development is shifting its attention [1] from “everything is an object”, so trendy in the 1980s and 1990s, to “everything is a model”.

ATRIUM [12, 15] (Architecture generaTEd from RequIREments applying a Unified Methodology) has been defined following the MDD principles, as models drive its application, and the tool MORPHEUS (see [13] for demos) has been built to support its models and activities. This methodology has been defined to guide the concurrent definition of requirements and software architecture, paying special attention to the traceability between them. In this context, the support of MORPHEUS is a valuable asset allowing the definition of the different models; maintaining traceability among them; supporting the necessary transformation, etc. This chapter focuses on MORPHEUS and its support to an MDD process.

This chapter is structured as follows: After this introduction, a brief description of ATRIUM is presented in Section 2. Section 3 describes the supporting tool of ATRIUM, MORPHEUS. Related works are described in Section 4. Finally, Section 5 ends this chapter by presenting the conclusions and further works.

2 ATRIUM at a Glance

ATRIUM provides the analyst with guidance, along an iterative process, from an initial set of user/system needs until the instantiation of the proto-architecture. ATRIUM entails three activities to be iterated over in order to define and refine different models and allow the analyst to reason about partial views of both requirements and architecture. Figure 1 shows, using SPEM [21] (Software Process Engineering Metamodel), the ATRIUM activities that are described as follows:

- *Modelling Requirements*: This activity allows the analyst to identify and specify the requirements of the system-to-be by using the *ATRIUM goal model* [18], which is based on KAOS [5] (Knowledge Acquisition in auTOMated Specification) and the NFR (Non-Functional Requirements) Framework [2]. This activity uses as input both an informal description of the requirements stated by the stakeholders and the CD 25010.2 *Software product Quality Requirements and Evaluation Quality model (SQuaRE* [9]). The latter is used as framework of concerns for the system-to-be. In addition, the architectural style to be applied is selected during this activity [15].
- *Modelling Scenarios*: This activity focuses on the specification of the *ATRIUM scenario model*, that is, the set of *Architectural Scenarios* that describe the system’s behaviour under certain *operationalization* decisions [16]. Each ATRIUM Scenario identifies the architectural and environmental elements that interact to satisfy specific requirements and their level of responsibility.
- *Synthesize and Transform*: This activity has been defined to generate the proto-architecture of the specific system [14]. It synthesizes the architectural elements from the ATRIUM scenario model that build up the system-to-be, along with its structure. This proto-architecture is a first draft of the final description of the system that can be refined in a later stage of the software development process.

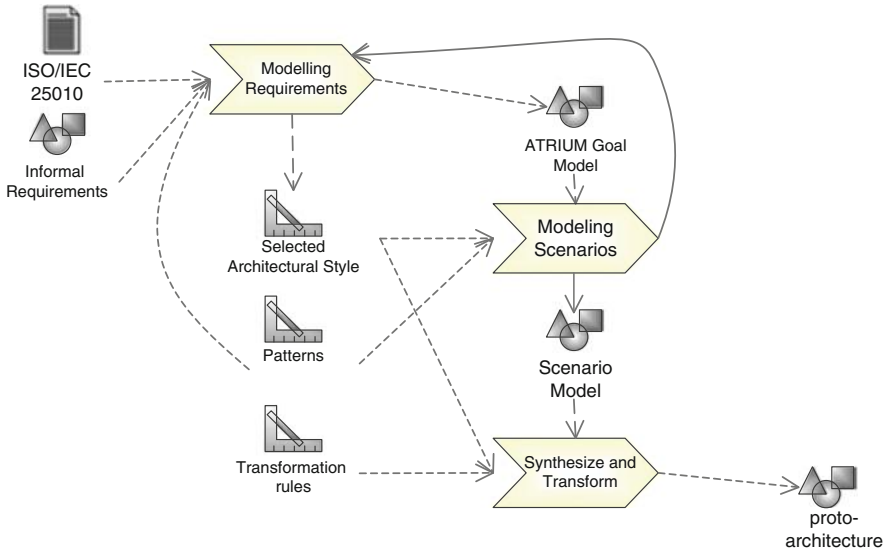


Fig. 1 An outline of ATRIUM

This activity has been defined by applying *Model-To-Model Transformation* (M2M, [4]) techniques, specifically using the QVT Relations language [20] to define the necessary transformations. It must be pointed out that ATRIUM is independent of the architectural metamodel used to describe the proto-architecture, because the analyst only has to describe the needed transformations to instantiate the architectural metamodel he/she deems appropriate. Currently, the transformations [15] to generate the proto-architecture, instantiating the PRISMA architectural model [22], have been defined. PRISMA was selected because a code compiler exists for this model.

ATRIUM has been validated in the context of the tele-operated systems. Specifically, the EFTCoR [8] project has been used for validation purposes. The main concern of this project was the development of a tele-operated platform for non-pollutant hull ship maintenance operations whose main structure is shown in Fig. 2. In this chapter, we are going to use the specification mode of the Robotic Device Control Unit (RDCU) to show how MORPHEUS provides support to each activity of ATRIUM. The RDCU is in charge of commanding and controlling in a coordinated way the positioning of devices along with the tools attached to them.

3 MORPHEUS: An MDD Supporting Tool

The main idea behind MORPHEUS is to facilitate a graphical environment for the description of the three models used by ATRIUM (goal model, scenarios model, and PRISMA model) in order to provide the analysts with an improved

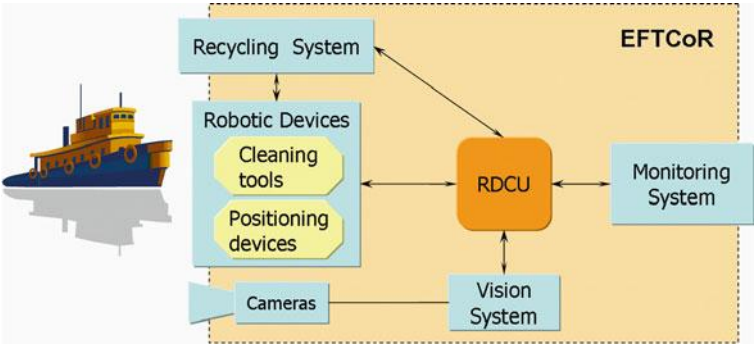


Fig. 2 Describing the EFTCoR platform

Fig. 3 Main architecture of MORPHEUS



legibility and comprehension. Several alternatives were evaluated such as the definition of profiles and the use of metamodeling tools. Eventually, we developed our own tool in order to provide proper integration and traceability between the models.

Figure 3 shows the main elements of MORPHEUS. The *Back-End* layer allows the analyst to access the different environments and to manage the projects he/she creates. Beneath this layer, the different environments of MORPHEUS are shown, providing each one of them support to a different activity of ATRIUM. The *RepositoryManager* layer is in charge of providing the different environments with access to the repository where the different models and metamodels are stored. In addition, each one of the graphical environments (Requirements Model Editor, Scenario Editor, and Architecture Model Editor) exploits Microsoft Office Visio Drawing Control 2003 [25] (VisioOCX in Figs. 4, 9, and 13) for graphical support. This control was selected to support the graphical modelling needs of MORPHEUS because it allows a straightforward management, for both using and modifying shapes. This feature is highly relevant for our purposes because all the kinds of concepts that are included in our metamodels can easily have different shapes, facilitating the legibility of the models. In addition, the user is provided with all the functionalities that Visio has, that is, she/he can manage different diagrams to properly organize the specification, zoom to see more clearly details, print the active diagram, etc. In the following sections, each one of the identified environments is described.

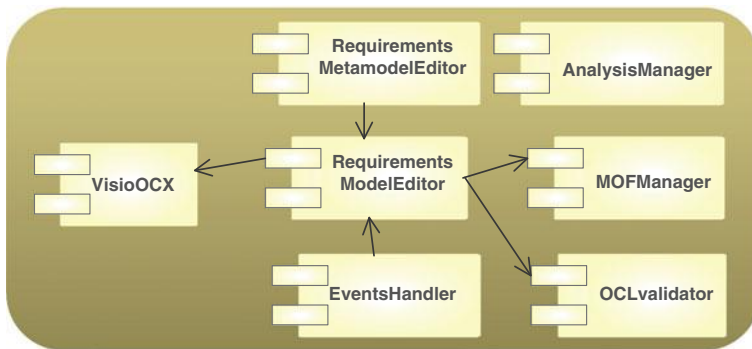


Fig. 4 Main elements of the requirements environment

3.1 Requirement Environment

As described in Section 2, *Modelling Requirements* is the first activity of ATRIUM. In order to support this activity, the *Requirements Environment* was developed. From the very beginning of the EFTCoR project, one of the main problems we faced was how the requirements metamodel had to change to be adapted to the specific needs of the project. With this aim, this environment was developed with two different work contexts. The first context is the *Requirements Metamodel Editor* (RMME), shown in Fig. 4, which provides users with facilities for describing requirement metamodels customized according to project's semantic needs (see Fig. 5). The second context is the *Requirements Model Editor* (RME), also shown in Fig. 4, which automatically offers the user facilities to graphically specify models according to the active metamodel (see Fig. 8). These facilities are very useful to exploit MORPHEUS to support other proposals.

It can be observed in Fig. 5 that the RMME allows the user to describe new meta-elements by extending the core metamodel described in Fig. 6, that is, new types of artefacts, dependencies, and refinements. This metamodel was identified and its applicability was evaluated by analysing the existing proposals in requirements engineering [18]. For instance, Fig. 5 shows that the two meta-artefacts (goal, requirement) of the ATRIUM goal model were defined using the RMME. In order to fully describe the new meta-elements, the user can describe their meta-attributes and the OCL constraints he/she needs to check any property he/she deems appropriate. Figure 7 shows how the meta-artefact *goal* was defined by extending *artifact*; describing its meta-attributes *priority*, *author*, *stakeholder*, etc.; and specifying two constraints to determine that the meta-attributes *stakeholder* and *author* cannot be null.

It is worth noting that automatic support is provided by the environment for the evolution of the model, that is, as the metamodel is modified, the model is updated in an automatic way to support those changes, asking the user to confirm the necessary actions whenever a delete operation is performed on meta-elements or

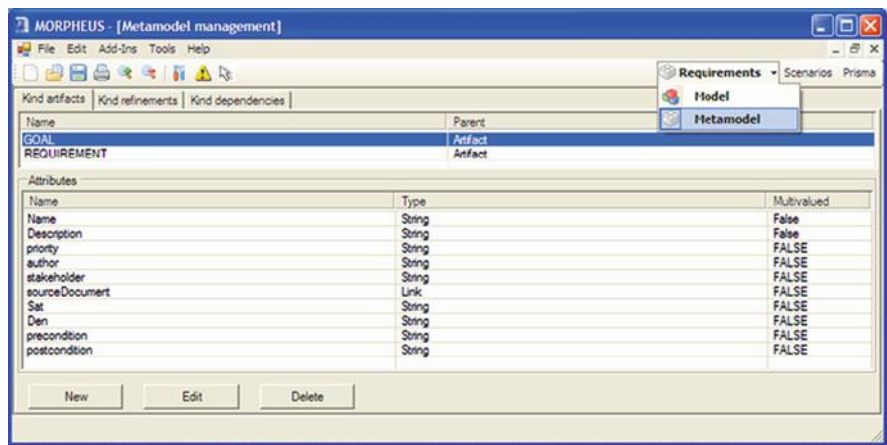


Fig. 5 Metamodelling work context (RMME) of the MORPHEUS requirements environment

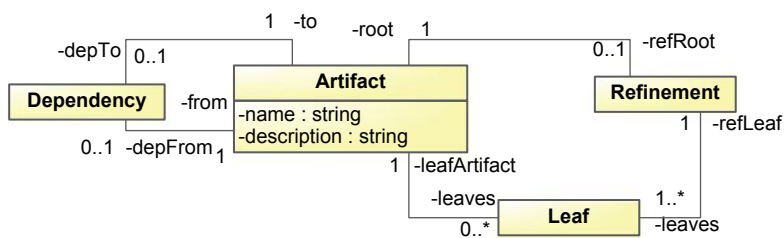


Fig. 6 Core metamodel for the requirements environment

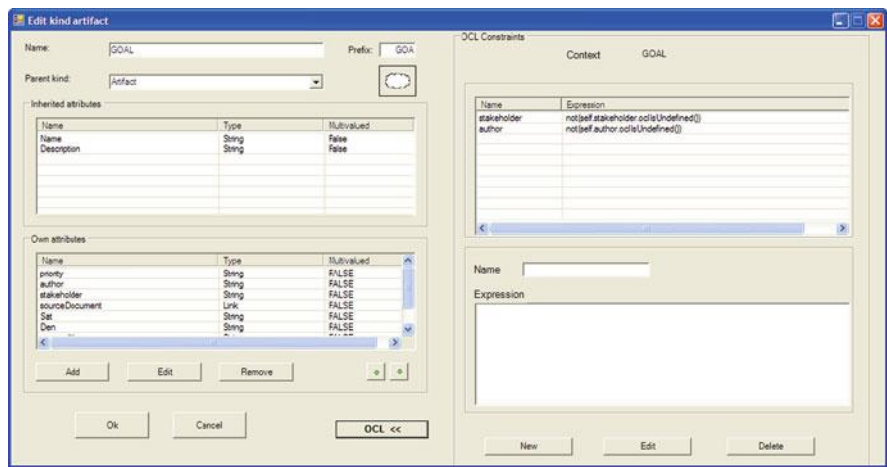


Fig. 7 Describing a new meta-artefact in MORPHEUS

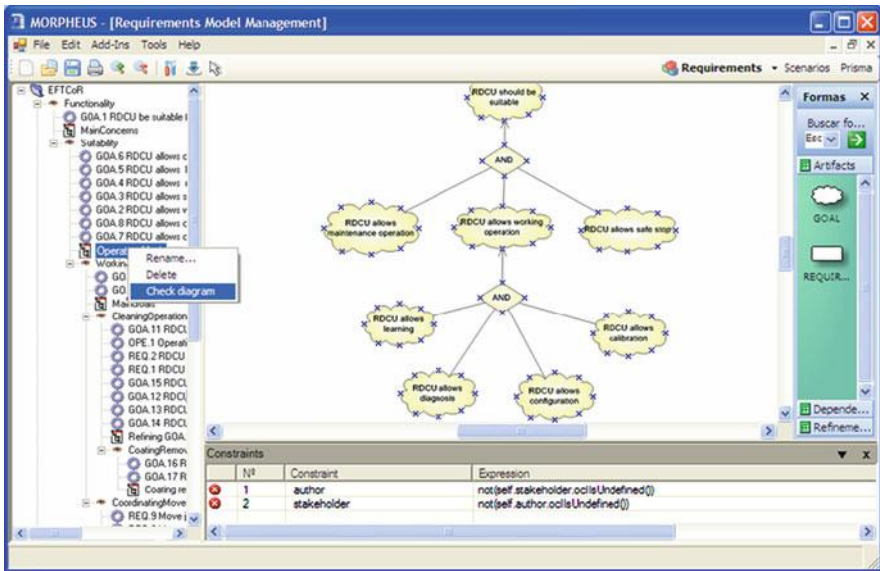


Fig. 8 Modelling work context (RME) of the MORPHEUS requirements environment

meta-attributes. This characteristic is quite helpful because the requirement model can be evolved as the expressiveness needs of the project do.

Once the metamodel has been defined the user can exploit it in the modelling context, RME, shown in Fig. 8. It uses VisioOCX to provide graphical support, as Fig. 4 shows, and has been structured in three main areas. On the right side, the *stencils* allow the user to gain access to the active metamodel. Only by dragging and dropping these meta-elements on the drawing surface in the centre of the environment, the user can specify the requirements model. He/she can modify or delete these elements by clicking just as usual in other graphical environments. For instance, some of the identified goals and requirements of the EFTCOR are described in the centre of Fig. 8. On the left side of the RME, a browser allows the analyst to navigate throughout the model and modify it. As Fig. 4 illustrates, the *EventHandler* is in charge of manipulating the different events that arise when the user is working on the RME.

In addition, as Fig. 4 illustrates, the RME uses two components to provide support to OCL: *OCLValidator* and *MOFManager*. The former is an engine to check OCL constraints which was integrated in MORPHEUS. The latter was developed to allow us to manipulate metamodels and models in MOF [19] format. By exploiting these components the constraints defined at the metamodel can be automatically checked. For instance, when the active diagram was checked, two inconsistencies were found that are shown at the bottom of Fig. 8.

However, the support of the tool would be quite limited if it only provides graphical notation. For this reason, the *Analysis Manager*, shown in Fig. 4, has been

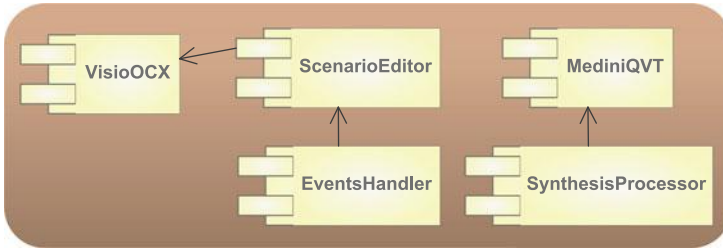


Fig. 9 Main elements of the scenario environment

developed to allow the user to describe and apply those rules necessary to analyse its models. These rules are defined by describing how the meta-attributes of the meta-artefacts are going to be valued depending on the meta-attributes of the meta-artefacts they are related to by means of meta-relationships. Once these rules are defined, the *Analysis Manager* exploits them by propagating the values from the leaves to the roots of the model [17]. This feature can be used for several issues such as satisfaction propagation [17], change propagation, or analysis of architectural alternatives [15].

3.2 Scenario Environment

As presented in Section 2, *Modelling Scenarios* is the next activity of ATRIUM. This activity is in charge of describing the scenario model. This model is exploited to realize the established requirements in the goal model by describing partial views of the architecture, where only shallow components, shallow connectors, and shallow systems are described. In order to describe these scenarios, an extension of the UML2 sequence diagram has been carried out to provide the necessary expressiveness for modelling these architectural elements [15]. In order to provide support to this activity the *Scenario Editor* (SME), shown in Fig. 10, was developed. The *ScenariosEditor* uses the VisioOCX to provide the user with graphical support for modelling the scenario model. The *EventHandler* is in charge of managing all the events triggered by user actions. Figure 10 illustrates how the SME has been designed. In a similar way to the RMME described in the previous section, it has been structured in three main areas. The *Model Explorer*, on the right, facilitates the navigation through the scenario model being defined in an easy and intuitive way and manages (creation, modification, and deletion) the defined scenarios. It is pre-loaded with part of the information of the requirements model being defined. For this reason, the selected *operationalizations*, catalogued by their dimensions, are displayed. It facilitates to maintain the traceability between the goal model and the scenario model. Associated to each operationalization one or several scenarios can be specified to describe how the shallow architectural elements collaborate to realize that operationalization. In the middle of the environment is situated the *Graphical View*

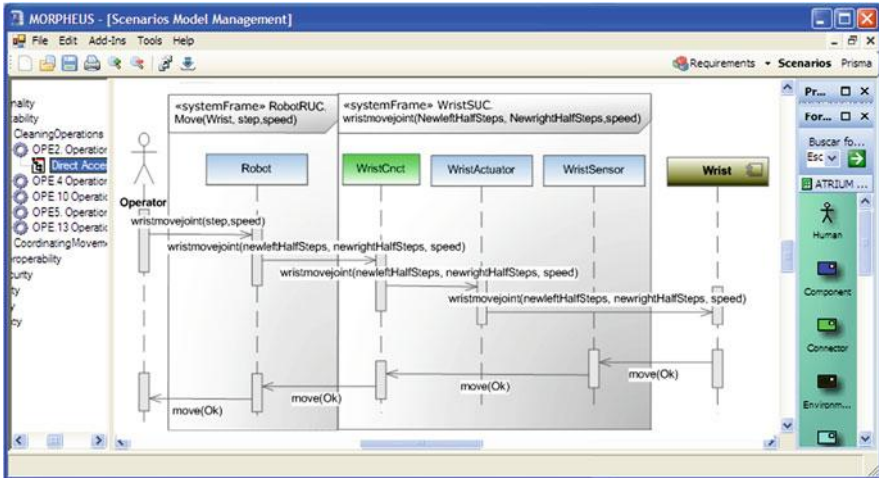


Fig. 10 What the Scenario Editor (SME) looks like

where the elements of the scenarios can be graphically specified. In this case, Fig. 10 depicts the scenario “OpenTool” that is realizing one of the operationalizations of the goal model. It can be observed how several architectural and environmental elements are collaborating by means of a sequence of messages. On the right side can be seen the *Stencil* that makes available the different shapes to graphically describe the ATRIUM scenarios. The user only has to drag and drop on the *Graphical View* the necessary shapes. In addition, below the stencil a control allows the user to introduce the necessary properties for each element being defined.

Another component of the scenario environment is the *Synthesis processor* (see Fig. 11). It provides support to the third activity of ATRIUM *Synthesis and Transform* which is in charge of the generation of the proto-architecture. For its development, the alternative selected was the integration of one of the existing M2M transformation engines considering that it has to provide support to the QVT Relations language. Specifically, a custom tool based in the *medini QVT* [11] transformation engine (licenced under the *Eclipse Public License*) was integrated as Fig. 11 illustrates. It accepts as inputs the metamodels and their corresponding models in XMI format to perform the transformation. This engine is invoked by the *Synthesis processor* which proceeds in several steps. First, it stores the scenario model being defined in XMI. Second, it provides the user with a graphical control to select the destination target architectural model, the QVT transformation to be used, and the name of the proto-architecture to be generated. By default, PRISMA is the selected target architectural model because the QVT rules [15] for its generation have been defined. However, the user can define its own rules and architectural metamodels to synthesize the scenario model. Finally, the *Synthesis processor* performs the transformation by invoking the QVT engine. The result is an XMI file describing the proto-architecture.

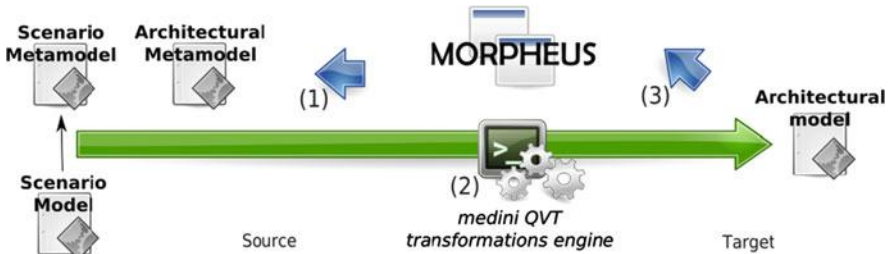


Fig. 11 Describing the Synthesis processor

3.3 Software Architecture Environment

As can be observed, both the requirements environment and the scenario environment provide support to the three activities of ATRIUM. However, as specified in Section 2, a proto-architecture is obtained at the end of its application. This proto-architecture should be refined in a later stage of development to provide a whole description of the system-to-be. With this aim the *Software Architecture Environment* [23] was developed. It makes available a whole graphical environment for the *PRISMA Architecture Description Language* [22] so that the proto-architecture obtained from the scenarios model can be refined.

As Fig. 13 depicts, this environment integrates VisioOCX for graphical support in a similar way to the previous ones. The *Architectural Model Editor* is the component that provides the graphical support, whose appearance can be seen in Fig. 12. It has three main areas: the stencil on the right where the PRISMA concepts are available to the user, the graphical view in the centre where the different architectural elements are described, and the model explorer on the right. It is worthy of note that this browser is structured in two levels following the recommendation of the ADL [23]: definition level, where the PRISMA types are defined, and configuration level, where the software architecture is configured.

As this environment should allow the user to refine the proto-architecture obtained from the synthesis of the scenario model, it provides her/him with facilities to load the generated proto-architecture if PRISMA was the selected target architectural model. In addition, it also provides an add-in that facilitates the generation of a textual PRISMA specification, which can be used to generate C# code by using the PRISMA framework.

4 Related Works

Nowadays, MDD is an approach that is gaining more and more followers in the software development area, and lots of tools that support this trend have arisen. Nevertheless, none of the existing solutions can completely cover the capabilities of the MORPHEUS tool.

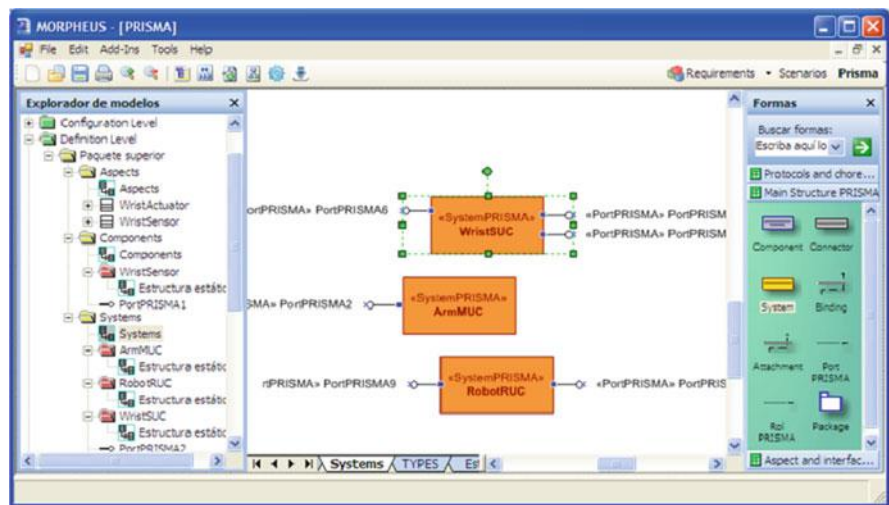
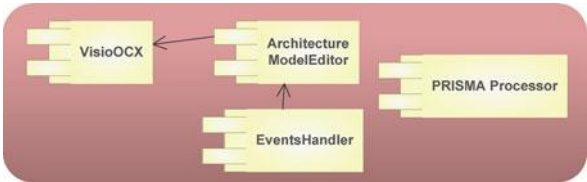


Fig. 12 What Architectural Editor looks like

Fig. 13 Main elements of the Software Architecture Environment



The Eclipse Modeling Framework (EMF) has become one of the most used frameworks to develop model-based applications. EMF provides a metamodeling language, called Ecore, that can be seen as an implementation of the Essential MOF language. Around EMF lots of related projects have grown that complement its modelling and metamodeling capabilities, such as OCL interpreters, model transformation engines, or even tools able to automatically generate graphical editors, such as Graphical Modeling Framework (GMF [7]). The advantages are twofold: first they are usually quite mature tools and second it is easy to interoperate with them by means of the XMI format. That is why the MORPHEUS tool has the *MOFManager* component: it allows us to reuse these tools as is the case of the OCL checker and the model transformations engine. Nevertheless, a solution completely based in EMF has also some important drawbacks. The main one is that, although it is not mandatory, this framework and its associated tools are fundamentally designed to deal with static models that do not change at run time. This factor makes frameworks like GMF completely useless for our purposes, because in MORPHEUS the requirements metamodel is populated with instances during its evolution and it is necessary to be able to synchronize them.

Other analysed alternatives are the MS DSL tools [3]. MS DSL tools are a powerful workbench that also provides modelling and metamodelling capabilities to automatically generate both code and graphical editors in Visual Studio. However, it exhibits the same weakness than the previous solution: it is basically designed to deal with models that do not evolve during time, so that these models can only be modified during design time and not at run time. Moreover, it lacks the wide community that provides complementary tools to deal, check, and analyse models, in comparison with the solution that is completely based on EMF. This disadvantage is also present in other tools, like the ones associated to Meta-CASE and Domain-Specific Modelling techniques, such as MetaEdit+ [10].

5 Conclusions and Further Works

In this work a tool called MORPHEUS has been presented, paying special attention to how it provides support to an MDD process, ATRIUM. It has been shown how each model can be described by using this tool and, specially, how traceability throughout its application is properly maintained. It is also worth noting the metamodelling capabilities it has, providing automatic support to evolve the model as the metamodel is changed. The integration of an OCL checker is also interesting as it allows the user to evaluate the model using the properties he/she deems appropriate.

Several works constitute our future challenges. Although the tool is quite mature, we are considering the development of other functionalities, as for instance, a model checker of the software architecture or a report generator for the requirements environment. It is also among our priorities to deploy the tool in the next future as an open source project to be evaluated and used by the community.

Acknowledgements This work is funded by the Department of Science and Technology (Spain) I+D+I, META project TIN2006-15175-C05-01 and by the UCLM, project MDDRehab TC20091111. This work is also supported by the FPU fellowship programme from the Spanish government AP2006-00690.

References

1. Bézivin J. (2004). *In search of a basic principle for model driven engineering*, Upgrade 5(2), pp. 21–24, 2004.
2. Chung L., Nixon B.A., Yu E. and Mylopoulos J. (2000). *Non-Functional Requirements in Software Engineering*, Kluwer, Boston.
3. Cook S., Jones G., Kent S. and Cameron A. (2007) *Domain-specific Development with Visual Studio DSL Tools*, Addison Wesley Professional.
4. Czarnecki K. and Helsen S. (2006) *Classification of Model Transformation Approaches*. IBM Systems Journal, 45(3), pp. 621–645.
5. Dardenne A., van Lamsweerde A. and Fickas S. (1993) *Goal-directed Requirements Acquisition*, Science of Computer Programming, 20(1–2), pp. 3–50.
6. *Eclipse Modeling Framework*. <http://www.eclipse.org/emf/>
7. *Eclipse Graphical Modeling Framework*. <http://www.eclipse.org/gmf/>

8. GROWTH G3RD-CT-00794 (2003) *EFTCOR: Environmental Friendly and cost-effective Technology for Coating Removal*. European Project, 5th Framework Prog.
9. ISO/IEC JTC 1/SC7 N4098 (2008), *Software Engineering-Software Product Quality Requirements and Evaluation (SQuaRE) Quality Model*.
10. Kelly S., Lyytinen K. and Rossi M. METAEDIT+ A fully configurable Multi-User and Multi-tool CASE and CAME Environment. *Proc. of 8th International Conference on Advances Information System Engineering*, LNCS1080, Springer, 1996, pp. 1–21.
11. Medini QVT, <http://projects.ikv.de/qvt>.
12. Montero F. and Navarro E. (2007), *ATRIUM: Software Architecture Driven by Requirements*, *Proc. 14th IEEE Int. Conf. on Engineering of Complex Computer Systems (ICECCS'09)*, IEEE Press, June 2007, in press.
13. MORPHEUS (2009), http://www.dsi.uclm.es/personal/ElenaNavarro/research_atrium.htm
14. Navarro E. and Cuesta C.E. (2008), *Automating the Trace of Architectural Design Decisions and Rationales Using a MDD Approach*, *Proc. 2nd European Conference Software Architecture*, LNCS 5292, Springer, September 2008, pp. 114–130.
15. Navarro E. (2007), *Architecture Traced from Requirements applying a Unified Methodology*, PhD thesis, Computing Systems Department, UCLM.
16. Navarro E., Letelier P. and Ramos I. (2007), *Requirements and Scenarios: Playing Aspect Oriented Software Architectures*, *Proc. 6th IEEE/IFIP Conf. on Software Architecture*, IEEE Press, 2007, n. 23.
17. Navarro E., Letelier P., Reolid D. and Ramos I. (2007) *Configurable Satisfiability Propagation for Goal Models using Dynamic Compilation Techniques*, *Proc. Information Systems and Development (ISD'07)*, Springer, New York, USA, September 2007, pp. 167–179.
18. Navarro E., Letelier P., Mocholí J.A. and Ramos I. *A Metamodeling Approach for Requirements Specification*. *Journal of Computer Information Systems*, 2006, 47(5), 67–77.
19. OMG (2006). Meta Object Facility (MOF) 2.0 Core Specification (ptc/06-01-01).
20. OMG (2005). Document ptc/05-11-01, QVT, MOF Query/Views/Transformations.
21. OMG (2005). Software Process Engineering Metamodel (SPEM), ver. 1.1 formal/05-01-06.
22. Pérez, J., Ali, N., Carsí, J.Á. and Ramos, I. (2006). *Designing Software Architectures with an Aspect-Oriented Architecture Description Language*, *Proc. 9th Int. Sym. on Component-Based Software Engineering (CBSE 2006)*, June 2006, pp. 123–138, Springer, Berlin/Heidelberg.
23. Pérez, J., Navarro, E., Letelier, P. and Ramos, I. (2006). *A Modelling Proposal for Aspect-Oriented Software Architectures*, *Proc. 13th Annual IEEE Int. Conf. and Works. on the Engineering of Computer Based Systems (ECBS'06)*, IEEE Press, March 2006, pp. 32–41.
24. Selic, B. (2003). *The Pragmatics of Model-Driven Development*. *IEEE Soft.* 20(5), pp. 19–25.
25. Visio 2003 (2009), [http://msdn.microsoft.com/en-us/library/aa173161\(office.11\).aspx](http://msdn.microsoft.com/en-us/library/aa173161(office.11).aspx)

Towards a Model-Driven Approach to Information System Evolution

Mohammed Aboulsamh and Jim Davies

Abstract Models have always played an important role in information systems (IS) design: typically, entity–relationship diagrams or object models have been used to describe data structures and the relationships between them. Model transformation and code generation technologies have given models an even more important role: as part of the source code for the system. This “model-driven” approach, however, has application beyond initial implementation. This chapter shows how subsequent changes to a design, captured as an “evolution model”, can be used to generate the data transformations required for the migration of data between different versions of the same system. The intention is to facilitate the adaptation of systems to changing requirements, using model-driven technologies for the rapid development of new versions, by reducing the cost and increasing the reliability of each migration step.

Keywords Model-driven · Information systems · Data migration · model evolution

1 Introduction

Model-driven software engineering [1] is the use of abstract models of structure and intended behaviour to generate or configure aspects of software systems. It is a generalisation of the *Model-Driven Architecture (MDA)* approach, characterised as “using modelling languages as programming languages” [2]. The automatic production of artefacts – source code, forms, services, and data – whose form is fully determined by a model can greatly reduce the amount of programmer or developer time required.

As well as reductions in cost, we should expect to see increases in quality and adaptability: the changes in requirements that follow initial development and deployment can be addressed in the design of a second version of the system – much

M. Aboulsamh (✉)
Oxford University Computing Laboratory, Oxford, UK
e-mail: mohammed.aboulsamh@comlab.ox.ac.uk

of which can be produced automatically, simply by making the necessary changes to the underlying models. This process of iterative, model-driven design and deployment could feasibly continue for the whole lifetime of a system.

However, where the system under development is an *information system* – a computing system intended for the collection and provision of data – deployment is likely to involve the creation or incorporation of large amounts of data within the system. This data may be sensitive or critically important to business processes; furthermore, complex relationships may exist between different items of data. The process of *data migration* is thus a key aspect of the deployment of a new version of an information system: we must ensure that existing data is properly transferred, and that any complex relationships are maintained.

The existence of detailed, faithful models of the two versions of the system can greatly facilitate this process. By comparing these models, we can determine immediately which data items have been added or removed; we can detect also changes in the way that a data item is used – in constraints, in the definition of operations or services, or in the description of interfaces and forms. By providing a suitable account of the changes made, we may be able to generate the transformations needed to effect the data migration.

In this chapter, we explore what would be required of a “suitable account”: how much information do we need to provide, if the data is to follow on automatically as the system evolves and in what form? We show how a language of basic change or evolution operations can be derived from the languages that define the models themselves, and how it may be extended to describe specific patterns of model evolution. We demonstrate how this language may be mapped to data migration programmes in existing model transformation frameworks.

This chapter begins with a formalisation of the intended approach and an account of how it may be applied to object models written in the Unified Modeling Language (UML). In Section 3, we show how the resulting language of model changes can be given an executable interpretation using the Eclipse-based ATLAS Model Management Architecture (AMMA) [3]. This chapter ends with a discussion of related and future work.

2 Object Model Evolution

Although information systems are still being described using entity–relationship diagrams [4, 5] or other techniques, object modelling is rapidly becoming the de facto standard in information systems design [6, 7]: it offers semantic support for classification, association, and specification of constraints and is supported by a wide range of sophisticated tools and implementation languages.

In the Unified Modeling Language (UML) [8], which may be considered representative of object modelling notations, a typical model will comprise a class diagram, some use case diagrams, and some sequence or state diagrams. Of these, the class diagram is the most likely to be used in the model-driven engineering of information systems.

The use case diagrams have little formal semantic content, and their importance lies in building an informal understanding of key aspects of the design. Sequence diagrams have more formal content, identifying sets of sequences of interactions that might be required or proscribed; however, as they provide only *examples* of what should or should not be achieved, their application – in a model-driven context – will usually be limited to test generation.

State diagrams define complete patterns of behaviour and thus may be used for software generation or configuration. In other domains, this has been done with considerable success: much of the programme code in embedded systems (such as car braking systems) is generated automatically from state diagram descriptions, with parameters derived from physical models.

However, for any aspect of system behaviour to be successfully, constructively modelled using a state diagram, it must be largely data-independent: diagrams with more than $O(10)$ states are difficult for humans to comprehend and are thus unlikely to be fit for the modelling purpose. In information systems, behaviour is anything but data-independent: our principal interest is in the values of attributes, the results of queries, and the validity of proposed updates.

In UML, this information can be presented in a class diagram. We may extend the basic feature set of classes, attributes, and associations with constraints written in the Object Constraint Language (OCL) [9]: these constraints can describe not only the data integrity properties, but also the pre- and post-conditions of update and query operations. A description of the language features required, as a fragment of the UML metamodel, is presented in Fig. 1.

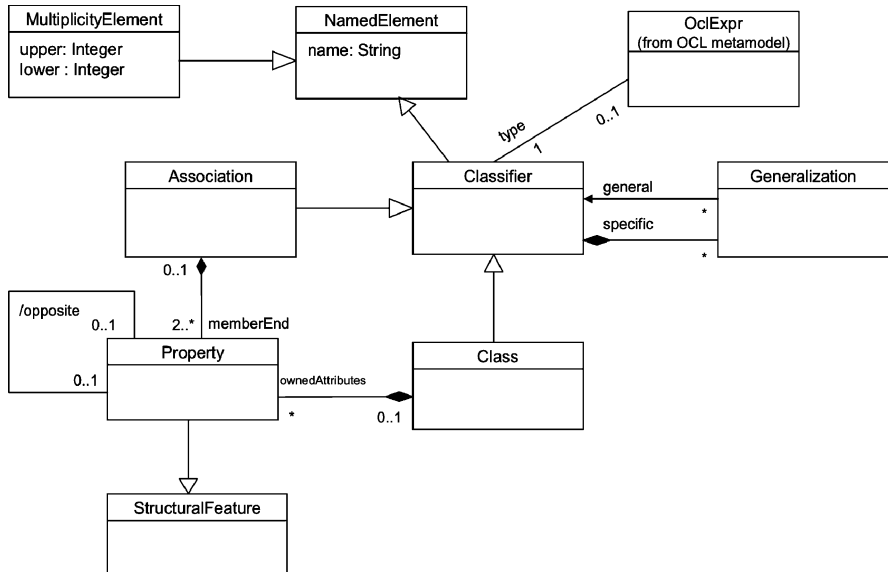


Fig. 1 Part of the UML2 metamodel

To adopt the terminology of the UML specification, the data held in a system is characterised as a *model instance*, an object at “modelling level” M0. If the system is in a valid state, then this data should conform to a *model* of that system, an object at level M1. This model should conform in turn to the description of the modelling language, a *metamodel*, an object at level M2.

Ideally, we might hope to devise a higher order function that would take any metamodel, any pair (before and after) of models conforming to that metamodel, and return a function that performs the corresponding transformation upon the data. If we were to name this function *migrate*, then it would have the type

$$\text{migrate: Metamodel}[X] \rightarrow \text{Model}[X] \rightarrow \text{Model}[X] \rightarrow \text{Data}[X] \rightarrow \text{Data}[X]$$

where X is a generic parameter identifying the modelling language (and thus the shape of models and data involved). However, it is easy to establish that an arbitrary pair of models, conforming to a given metamodel, will not necessarily provide the information that we require.

For example, consider the situation in which we delete an attribute from a class and introduce a new attribute of the same type. Inspection of the models before and after is not enough to tell us whether the data collected against the deleted attribute should be automatically migrated and stored against the new one. Accordingly, we aim instead to define a function with the type

$$\text{migrate: Metamodel}[X] \rightarrow \text{Model}[X] \rightarrow \text{Evolution}[X] \rightarrow \text{Data}[X] \rightarrow \text{Data}[X]$$

where $\text{Evolution}[X]$ denotes the language of changes or editing operations that may be made to a model conforming to the given metamodel. The definition of this language of changes can be given as a metamodel, at level M2, as an extension of the metamodel used to define the modelling language involved.

In the particular case of the UML, the evolution metamodel required extends the UML metamodel with classes describing the addition, deletion, and modification of model features: classes, attributes, and associations. Subclasses describe modifications to particular kinds of associations, denoting generalisation, aggregation, or composition. Compound operations can be added to the metamodel to represent common patterns of model evolution: for example, the introduction of an association class or the movement of an attribute up or down an inheritance hierarchy. The literature on schema evolution is an appropriate source of candidate operation patterns: see for example [10–12].

A particular model edit or evolution step may not always be applicable: it may be that the structure of the model does not permit the change in question – that the resulting model would no longer conform to the language definition metamodel – or it may be that it would be impossible to migrate the current data to fit the new version of the system model. An example of the former would be the proposed introduction of a new class using the same name as another class already defined in the model. An example of the latter would be a modification to the multiplicity of an association from $*$ to 0.1, when there are two or more links made for that association in the existing data (and there is no additional information to tell us which of the links should be deleted). Accordingly, the data migration rules that correspond to our

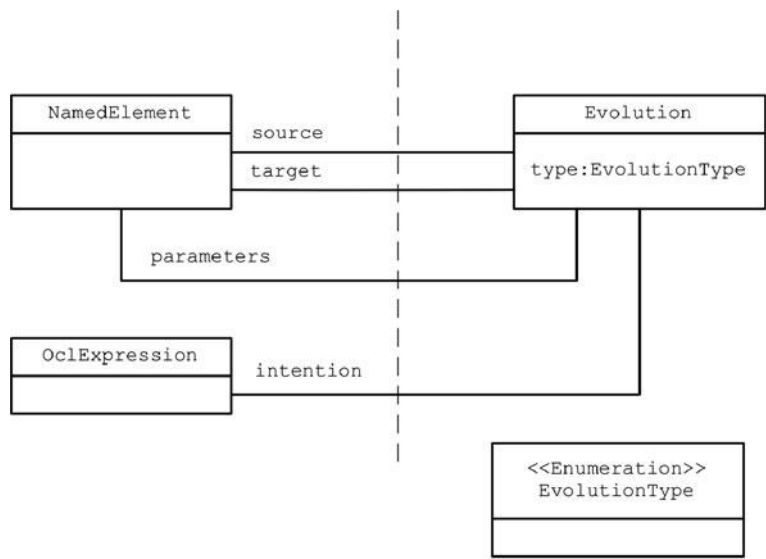


Fig. 2 Part of an evolution metamodel for UML

changes may have non-trivial *guards*, constraints upon the elements of the current model, or upon the current instance data.

Figure 2 shows part of an evolution metamodel for UML. Each evolution specification may refer to an arbitrary number of model elements in the source (current) and target (new) models and also to other model elements as parameters. Detailed intentions regarding the interpretation or re-use of new and existing data can be specified using OCL expressions. The classes to the left of the broken line are drawn from the UML and OCL metamodels: in general, any constraint or action language could be used provided that it is consistent with the definition of elements in the modelling language. The Evolution class can be subclassed to address different types of change, as described above: here, we have assumed only that a suitable enumeration can be provided.

The first class of constraints, those that are independent of the current instance data, and are derived from the constraints of the language metamodel, may be described as invariants in the context of Evolution: they will then be added in the production of every migration rule. For example,

```
context Evolution
inv classAddedIsUnique :
    self.evolutionType = EvolutionType :: classAdded implies
        self.target.Classifier→isUnique(Class)

    to represent the constraint mentioned above or
context Evolution
```

```

inv uniAssocAddedSrcIsnotTrgt :
  self.evolutionType = EvolutionType :: uniAssocAdded implies
    self.source.Classifier.Class→
      intersection(self.target.Classifier.Class)→isEmpty()

```

to express the requirement that a unidirectional association cannot be added if the proposed source and target classes are the same (such an association would, by its nature, be bidirectional, and hence its classification would be inconsistent).

The second class of constraints apply to the instance data, and these are derived from constraints in the system model, rather than the language metamodel. These constraints express data integrity properties and themselves fall into two subclasses: basic *representation* constraints, such as those of typing and referential integrity, and system-specific *invariants*, expressing the intended interpretation of – and intended relationships between – data values and links.

These will be included as OCL expressions within the system model: for example, the following constraint (in a model of an airline booking system) would express the requirement that no person should be both confirmed and on the waiting list for a given flight

```

context Flight
inv waitlist→intersection(confirmed)→isEmpty()

```

The addition of such a constraint to a model requires no data migration in itself; it does, however, require that the existing data, following whatever other changes are proposed, should satisfy this restriction: it leads to a non-trivial guard upon the overall data migration function.

These constraints become particularly important when we propose model edits that correspond to changes in interpretation – our account of the data semantics may itself evolve following initial deployment. For example, if we were to also change our interpretation of waitlist above so that it included those who had requested an “upgrade” to a higher class of service on the same flight, and indicate this, for example, through the expression

```
waitlist := waitlist→union(upgradelist)
```

then the resulting data migration operation, generated from the specification of the changes to the model, would be guarded by the constraint that none of those on the upgrade list should hold a confirmed seat for that flight.

Such changes of interpretation are inevitable and will typically involve more than one attribute or association. Our language of model edits needs to include a facility for annotating attributes and associations with expressions representing their new intended value – or, equivalently, simple assignments or substitutions, as in the above example. The corresponding transformations, considered in the context of whatever other changes are proposed at the same stage of evolution, can lead to guards that may be difficult to determine through manual inspection.

It is particularly important that the generated guards are correct. They will be used to determine whether or not the proposed data migration can proceed. If the guards are too weak, then the existing data, transformed according to the migration data function, might not conform to the new version of the model – violating integrity constraints at the model or metamodel level.

3 Implementation

To show how this approach may be developed and applied in the particular context of UML, consider the proposed model evolution shown in Fig. 3.

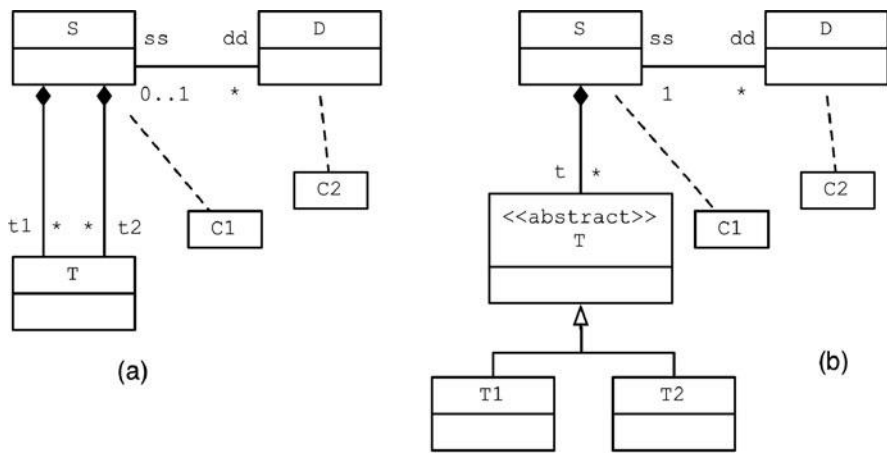


Fig. 3 Two versions of a model

In the first version of the model – shown in Fig. 3a – each instance of class S consists of two collections of instances t1 and t2 of another class T; there is also a bidirectional association between S and a third class D in which each instance ss of S may (or may not) correspond to a collection of instances dd of D. The fact that this is a bidirectional association and not simply a pair of associations is enforced by the pair of OCL constraints C1 and C2,

```
context S inv C1 : dd→forall(dx | dx.ss = self)
context D inv C2 : ss.dd→includes(self)
requiring that the ss and dd roles correspond in the expected fashion.
```

In the second version – shown in Fig. 3b – class T has been specialised into two subclasses T1 and T2 and is now used as an abstract class. In addition, the multiplicity of the association between S and D has been modified: as a property of D, it is now mandatory, rather than optional. The implications for data migration should be obvious: every instance of T needs to be assigned either T1 or T2 as a type, and every instance of D now needs to be associated with exactly one instance of S.

These changes may be captured as a series of model edits or evolution steps against the UML metamodel, instances of Evolution in the sense of the metamodel of Fig. 2.

```

1 uniAssocModified(D, ss, S, 1)
2 classAbstracted(T)
3 classAdded(T1)
4 generalizAdded(T1, T)
5 uniAssocDeleted(S, t1)
6 classAdded(T2)
7 generalizAdded(T2, T)
8 uniAssocDeleted(S, t2)
9 uniAssocAdded(S, t, T,*)

```

If our enumeration type, or our hierarchy of subclasses, permits, we may be able to combine a number of these changes into a single step, as an instance of an evolution pattern. For example, Steps 2–5 above might be combined into a single step of the class `SpecializeReferenceType`: the same pattern is repeated twice: once for Class T1 and again for Class T2. Furthermore, the operation of changing an association multiplicity from optional to mandatory has similar implications for data whenever it is applied, and we might usefully define another subclass or type to model this.

The value of identifying an evolution pattern and adding it as an explicit operation to the language will depend upon the context in which the model evolution takes place and the manner in which changes to intended interpretations are specified during the model editing process. In the case of the change from optional to mandatory above, it should be a simple matter for any model editor to determine that a sequence of editing operations on an association property has precisely this overall effect and to add the corresponding compound operation to a list of changes made; in other cases, the designer will need to spell out the intended effect of their changes in terms of the relationships between attributes and associations in the current and proposed versions of the model.

Each evolution step at the model level can be implemented as a series of data migration rules. For example, the `SpecializeReferenceType` operation could be implemented as

```

modifyAttributeType(this, t1, T1)
removeLink (this, t1, T)
insertLink (this, t, T)

```

iterated over all instances this of class S, and the operation of changing multiplicity from optional to mandatory could be implemented as a single guard:

```
this.ss→size() = 1
```

iterated over all existing instances this of class D, in the absence of any user-supplied expression identifying a default link to be made for any instance that currently lacks a partner object of class S for this property.

A prototypical implementation of this approach has been constructed using the Eclipse ATLAS Model Management Architecture (AMMA) [3]. This is an extensible, metamodel-based framework that supports *model weaving*: the specification of relationships between elements in different models. An extension of the Atlas Model Weaver (AMW) [13] weaving metamodel was used as the basis of an evolution metamodel, allowing model coding using the KM3 (Kernel Meta-Meta Model) [14] notation.

The Atlas Transformation Language (ATL) [15] was used to create higher order transformations – the ATL metamodel can itself be used as input to model transformations – that generate transformations, mapping operations at the model level into operations at the instance or data level. For example, the data migration rule shown in Fig. 4 would be generated to produce instances t of type $T1$ or $T2$ in the new model from instances $t1$ and $t2$ in the existing data.

```

1  module DataModel_instanceMigration;
2  create OUT : targetData_v02 from IN : sourceData_v01;
3  rule S {
    from c1: sourceData_v01!S
    to c2: targetData_v02!S (
      t <- c1.t1->collect(p | thisModule.T1(p))->
      union(c1.t2->collect(p | thisModule.T2(p)))
    )
4  }

5  lazy rule T1 {
    from p1: sourceData_v01!T
    to p2: targetData_v02!T1 (
      ...
    )
6  }
7  ...

```

Fig. 4 Part of an instance-level data transformation rule

4 Discussion

Large organisations regularly invest considerable time and effort in upgrading their information systems, hoping for an evolutionary progression from the existing provision to a new system or architecture that will better support their business practices or goals. In the process, the challenge of data migration is often recognised, but usually seen as very much a secondary concern – it is unlikely to have any significant influence upon the selection of a new system or the design of a new architecture. As a result, it is often left to the final, post-implementation stage of a project, sometimes with costly and unforeseen results [16]. The advent of model-driven engineering offers an opportunity to consider the question of data migration, in detail, at the design stage: a precise account of the proposed changes to the system model can be used to predict the consequences for existing data and to generate the necessary migration functions.

Although studies such as [17, 18] suggest that the majority of model evolution steps are quite straightforward – the addition of classes, attributes, or associations, the merging or splitting of classes, or the modification of the domain or cardinality of an attribute, existing model editing environments have done little to address the potential for automating the production of the migration functions that would follow such a change [19], often leaving valuable, critical data to be transformed and loaded by hand-written SQL scripts – long after the design process has been completed.

There is, however, a considerable body of related and applicable work. Ref. [20] proposes the automatic generation of a migration plan specified in terms of migration expressions and transformation modules. This work has similar goals to our own, but is realised in quite a different fashion, without the (higher order) use of the language metamodel and without the emphasis upon full automation: this is due in part to the focus upon the recovery of data from legacy, relational databases, where the approach set out here would present little or no advantage – instead of abstract, partial OCL expressions, any changes in interpretation would need to be rendered explicitly as SQL queries.

The work on database schema evolution – usually taken to mean the application of changes while the database is in operation [21] – is also relevant. Approaches such as those described in [10, 11, 22] offer support for evolutionary steps, but this support is implementation-specific [23], rather than being made available at the modelling language level.

Fowler [24] has promoted the notion of *refactoring* as means of revising models, and hence code, to improve the design without changing the externally observable behaviour. Of course, any change to an object model might be detectable in terms of the available or effect of some operation, but the general principle is clear: key operations should be unaffected. Ambler and Sadalage [25] has extended refactoring principles to database design and looked also at “live instances” of data. However, our concerns here are more general – we do not require evolutions to be “behaviour preserving” – and our emphasis is upon a model-driven, automated approach.

Cicchetti et al. [26] have proposed, in common with this chapter, capturing model changes as simple difference operations, conforming to a difference metamodel of additions, deletions, and modifications. In common with [27], such a difference-based approach admits automatic production of migration functions only in the simplest of cases: as we have demonstrated, an expression language is required to describe the intentions regarding data values and associations.

The Epsilon Comparison Language (ECL) [28] includes a rule-based metamodel-independent language, intended for model composition and model transformation testing. Existing work has focussed upon model comparison, but nothing concrete has been proposed as yet regarding data migration from descriptions of model evolution. The same is true of EMF Compare [29], included as part of the Eclipse Modeling Framework Technology (EMFT) project. However, there is clear potential for the integration of our approach with these technologies.

References

1. J. Bézuvin. On the unification power of models. *Software and Systems Modeling*, 4(2). 2005.
2. D.S. Frankel, *Model Driven Architecture*. Wiley 2003.
3. J. Bézuvin, F. Jouault, P. Rosenthal and P. Valduriez. Modeling in the large and modeling in the small. In *Proceedings of European MDA Workshops: Foundations and Applications*. Springer LNCS 3599. 2005.
4. P.P. Chen. The entity-relationship model – toward a unified view of data. *ACM Transactions on Database Systems*, 1(1). 1976.
5. B. Thalheim. *Fundamentals of Entity-relationship Modeling*. Springer. December 1999.
6. M. Blaha and W. Premerlani. *Object-Oriented Modeling and Design for Database Applications*. Prentice-Hall. 1998.
7. J. Davies, J. Welch, A. Cavarra and E.Crichton. On the generation of object databases using booster. In *Proceedings of the 11th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS)*. 2006.
8. Object Management Group (OMG). Unified Modeling Language (UML) Infrastructure, Version 2.2. <http://www.omg.org/docs/formal/09-02-04.pdf>. Accessed May 2009.
9. Object Management Group (OMG). OCL 2.0 Specifications, version 2. <http://www.omg.org/docs/ptc/05-06-06.pdf>. Accessed May 2009.
10. J. Banerjee, W. Kim, H. Kim and H. Korth. Semantics and implementation of schema evolution in object-oriented databases. *Proceedings of ACM SIGMOD 87*. ACM Press. 1987.
11. F. Ferrandina, T. Meyer, R. Zicari, G. Ferran and J. Madec. Schema and database evolution in the O2 object database system. In *Proceedings of the 21th International Conference on Very Large Data Bases (VLDB)*. 1995.
12. K. Claypool, J. Jin and E. Rundensteiner. SERF: schema evolution through an extensible, re-usable and flexible framework. In *Proceedings of International Conference on Information and Knowledge Management*, 1998.
13. M. Del Fabro, J. Bézuvin, F. Jouault, E. Breton and G. Gueltas. AMW: a generic model weaver. In *Proceedings of the 1ères Journées sur l'Ingénierie Dirigée par les Modèles*. 2005.
14. F. Jouault and J. Bézuvin. KM3: a DSL for Metamodel Specification. In *Proceedings of 8th IFIP International Conference on Formal Methods for Open Object-Based Distributed Systems*. Springer LNCS 4037. 2006.
15. F. Jouault and I. Kurtev. Transforming Models with ATL. In *Proceedings of MoDELS 2005*. Springer LNCS 3844. 2006.
16. T. Friedman. *Risks and Challenges in Data Migrations and Conversions*. Gartner Research. February 2009.
17. J. Hainaut, A. Cleve, J. Henrard and J. Hick. *Migration of Legacy Information Systems*. Software Evolution. Springer. 2008.
18. B. Bordbar, D. Draheim, M. Horn, I. Schulz and G. Weber. Integrated model-based software development, data access, and data migration. In *Proceedings of the 8th International Conference, MoDELS 2005*. Springer LNCS 3713. 2005.
19. T. Friedman, M. Beyer and A. Bitterer. Magic Quadrant for Data Integration Tools. Gartner Research. <http://mediaproducts.gartner.com/reprints/sas/vol5/article4/article4.html>. Accessed July 2009.
20. A. Boronat, J. Pérez, J.A. Carsí and I. Ramos. Two experiences in software dynamics. *Journal of Universal Computer Science*, 10(4) (2004), 428–453.
21. C. Delgado, J. Samos and M. Torres. Primitive operations for schema evolution in ODMG databases. In *Proceedings of the 9th International Conference on Object-Oriented Information Systems (OOIS)*. Springer LNCS 2817. 2003.
22. B. Lerner. A model for compound type changes encountered in schema evolution. *ACM Transactions on Database Systems*, 25(1). 2000.

23. A. Rashid, P. Sawyer and E. Pulvermueller. Flexible approach for instance adaptation during class versioning. In Proceedings of the 6th International Conference on Object Oriented Information Systems. Springer LNCS 1944. 2000.
24. M. Fowler. Refactoring: Improving the Design of Existing Code, Addison Wesley, 1999.
25. S.W. Ambler and P.J. Sadalage. Refactoring Databases: Evolutionary Database Design, Addison Wesley Professional, 2006.
26. A. Cicchetti, D. Di Ruscio and A. Pierantonio. A metamodel independent approach to difference representation. Journal of Object Technology, 6(9). 2007.
27. Y. Lin, J. Gray and F. Jouault. DSMDiff: A differentiation tool for domain-specific models. European Journal of Information Systems, 16 (4). 2007.
28. D. Kolovos, R. Paige and F. Polack. Model comparison: a foundation for model composition and model transformation testing. In Proceedings of the International Workshop on Global Integrated Model Management. ACM. 2006.
29. <http://www.eclipse.org/modeling/emft/?project=compare>. Accessed May 2009.

Open Design Architecture for Round Trip Engineering

Miroslav Beličák, Jaroslav Pokorný, and Karel Richta

Abstract This chapter introduces a component design of application logic in special type of information system architecture called Open Design Architecture for Round Trip Engineering (ODARTE). This architecture supports model-driven development and integrates information system design with its executable form. It is possible to extract design anytime, modify it, and load back to change the activity and behavior of information system. In this approach, the application logic represents solely functionality and can be described either by sequential model of Windows Workflow Foundation or by UML activity or interaction diagrams. This approach allows creation of flexible and modifiable meta-design of application logic. Finally, an experimental simulation is shown demonstrating the effect of proposal which relates to pilot version of runtime environment for ODARTE support.

Keywords MDA · Workflow · Component-based development · UML

1 Introduction

Information system architecture (ISA) exposes an important role in software engineering area besides information system (IS) development. On one hand we can consider architecture of a system as an organization structure of system including its parts: decomposition, connectivity, interactions, mechanisms, and directive axioms that come into system design [1], or top-level concept of system in its own environment. On the other hand, an architecture is an interception of strategic aspects of higher system structure and presents specific view on IS and its structure,

M. Beličák (✉)

Department of Software Engineering, Faculty of Mathematics and Physics, Charles University in Prague, Prague, Czech Republic
e-mail: miroslav.belicak@mff.cuni.cz

subsystems with their interactions, relations between IS and environment in which IS is deployed, processes and methodologies of development in IS area.

Therefore, the choice of ISA plays one of the key roles during IS production, because ISA provides both miscellaneous potentialities and constraints for software implementation. Also, such properties as robustness, modifiability, scalability, adaptability, flexibility for different changes of environment (in which IS is deployed) or quality of security infrastructure depend upon type of architecture.

The component-based architectures (CBA) are increasingly adopted and used by software engineers. Generally, we can consider CBA as an architecture which includes mechanisms and techniques for developing reusable software implementation units called components. A *component* defines its behavior and provides its functionality through provided and required interfaces. An *interface* is a kind of classifier that represents a declaration of a set of coherent public features and obligations and specifies a *contract*; any instance of a classifier that realizes the interface must fulfill that contract [3].

The chapter deals with our proposal of the special architecture, which we call the Open Design Architecture for “Round Trip Engineering” (ODARTE). Section 2 introduces our motivation; subsequently basic principles of ODARTE and its structure are described in Sections 3, 4, and 5. At the end some experimental results are presented in Section 6.

2 Our Motivation

The basic goal of our research is to eliminate problems related with dynamic evolution of IS. We can divide these problems into next two main spheres:

- problems on IS creator side which relate to development and changes of IS and
- problems on IS user side which relate to IS utilizing, maintenance, actualization, and new requirements.

Incomplete IS design and incomplete relationship between IS design and IS implementation – in case of incomplete design (either from time stress by IS development reason or from weaker designers’ abilities reason) – rise problems related to depressed binding between design and implementation. One thing is the design and another is an implementation. Basically, nothing restrains developers by deflect from design foundations and go by their own way anytime.

Other problems relate to new requirements of IS user. Impact of new requirements onto existing IS – the design of IS must be sufficiently flexible for additional changes impact reducing, especially by sensitive structures and parts of IS. The designers must take this fact into account [2].

It was our aim to create architecture of IS, which would eliminate main problems on user and creator side. User of IS will not be completely dependent only on IS creator. In the case of additional requirements for IS, it would be possible to realize

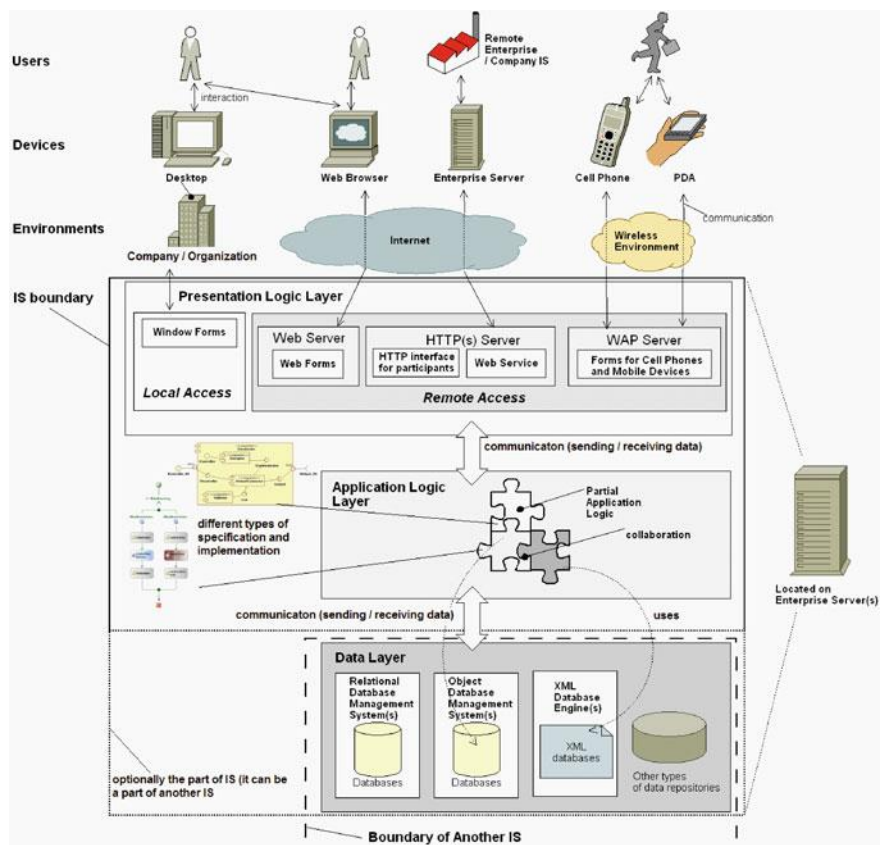


Fig. 1 Main layers of technological view of information system

these changes without IS creator. The basic problems are on application logic level. This level is the layer of IS which represents whole functionality of IS (Fig. 1). The application logic represents operations over processed data stored in data layer. The description of this application logic should be based on existing standards; we do not want to create some specific proprietary solution. Accordingly to wide usage of object-oriented and component approach, we decide for combination of standard component-based architecture and workflow-based architecture where description of application logic is realized by UML [17] and XOML [13].

For the possibility of easy changes, application logic should be totally transparent. We can imagine direct interpretation of source code without previous compiling. Such approach, in accordance to its restricted possibilities (which particularly relates with application and runtime performance), is basically unacceptable. This fact is expatiated by reality that some ISs are formed by a combination of different development platforms and technologies (such as .NET or Java). Therefore IS must have certain part which is compiled into executable form, either for machine

code for target machine or for machine language for virtual machine. Hereby, this is a constraint for later modification of IS, if we assume that we do not take disassembling and recompilation possibilities into account.

Currently best-known architectures of IS such as service-oriented architecture (SOA) [5], model-driven architecture (MDA) [10], component-based architecture [3], and mentioned workflow-based architecture (WBA) [14] partially solve this problem, each in their own way:

1. in the case of SOA there is a possibility of dynamic web services orchestration,
2. in the case of MDA we have platform-independent model at disposal,
3. in the case of CBA we can obtain (analogously by MDA) IS models which will be directly transformed into low-level form or into executable form (e.g., if IS was designed by Executable UML – xUML [13]), and

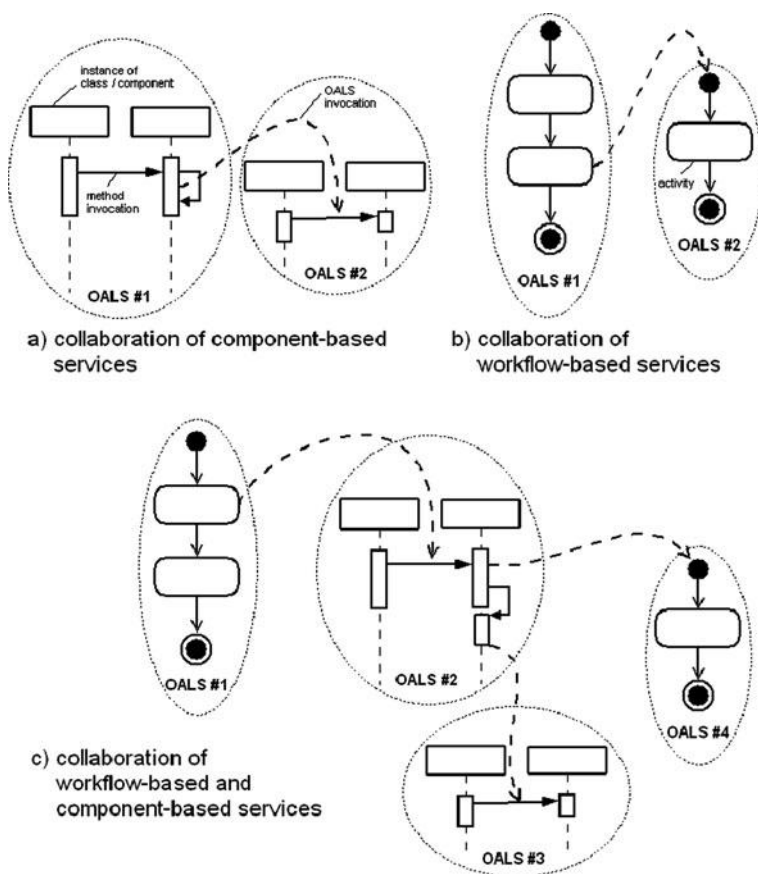


Fig. 2 Examples of service invocation combinations

4. in the case of WBA we have description of workflow (i.e., in XOML form) at disposal, which is in general a form understandable for designers of integrated development environment.

Common feature of the problem lies in the fact that executable (or low-level) parts of IS are in compiled form [7]. The only possibilities are disassembling and recompilation, but as was mentioned earlier, it requires enormous time amount. Similar solutions are SOA and WBA architectures, but in SOA there is the problem of granularity of web services. In WBA there is the resulting workflow compiled into executable form and appropriate formalism for workflows is missing. ODATE emulates in its services orchestration (Fig. 2) flexibility of web services and fine-grain description with formal specification of services on the base of workflows.

3 Basic Principles of ODATE

Our proposed ODATE architecture (Fig. 3) tries to solve mentioned problems. IS based on this type of architecture has so-called open design, which is integrated together with solely IS and thus they form one logical unit. There is a possibility to extract anytime this design from ODATE and subsequently modify it. In accordance to these modifications, the activity and behavior of IS are changed. The runtime environment that supports this architecture is responsible for activity and behavior changes. Later, the modified design of IS is affiliated to IS, which is this way ready for new eventual modification.

At this place, let us remark that “open part” of IS relates to the layer of IS’s application logic which represents solely functionality of IS. Under the term “application logic” we can here imagine different services which are provided for IS users or internally used by IS. These services can process data administered by IS, or different computations, network operations, connecting to different data repositories together with their utilizing, etc.

This way ODATE tries to find an appropriate compromise between two mentioned extremes: “total transparency” (e.g., direct interpreting of application from source code) and “total compiled form” (e.g., executable form without any reflection) and, also, ODATE does not want to divert from existing standards.

As was mentioned, nowadays it is not possible to have totally transparent IS or application. So, certain part must be in compiled, executable (very often binary) form. On the other hand, it is desirable to get certain meta-information about IS, therefore certain part of IS must be “readable” or understandable for specialists to utilize as a knowledge about IS. Therefore, it is necessary to determine certain compromise or ratio between such executable and readable-opened part of IS. As this ratio mostly depends upon ad hoc solved problem, ODATE allows IS designers to determine it directly. Opened design of ALS (application logic specification) must

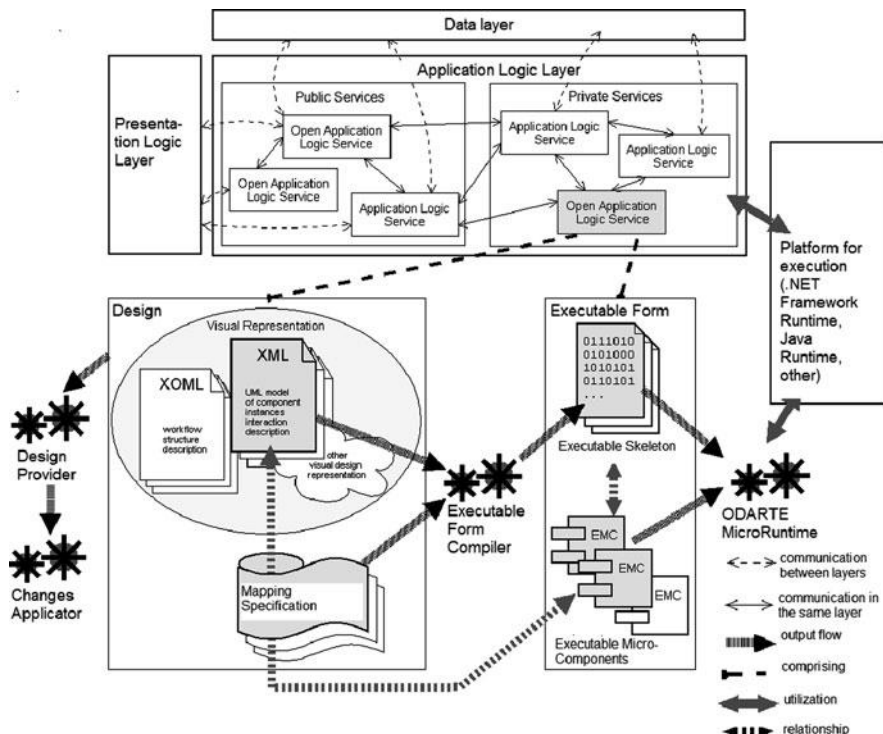


Fig. 3 Principle of Open Design Architecture for Round Trip Engineering

be readable by certain CASE tool or integrated development environment (IDE) in order to provide relevant information to designers. ODATE for such design description uses workflows based on Windows Workflow Foundation (shortly WF) or models of component instances interaction known from UML.

Executable functionality is hidden in so-called executable micro-components (EMCs). EMCs contain partial functionality of certain area of IS functionality and are responsible for this functionality. For example, if this functionality concerns with database operations, one EMC has to ensure connectivity with SQL server, other EMC ensures selection of data, another EMC ensures activation of certain transactions, etc. At the same time, one EMC can be used in multiple services, so from all the EMCs we can create multiple instances for performing in different services of application logic.

The “glue” between design and EMCs is created by so-called mapping specifications. There is necessity to take into account that service design in ODATE does not cover some functionality in principle, but only the way for description of its execution. Let us remind that the design deals with one part of UML model for objects interaction. Mapping specifications define what will be performed in particular activities of workflow.

So, the ALS which encapsulates design and mapping specifications and eventually EMC we call Open Application Logic Service (OALS). Mapping specification

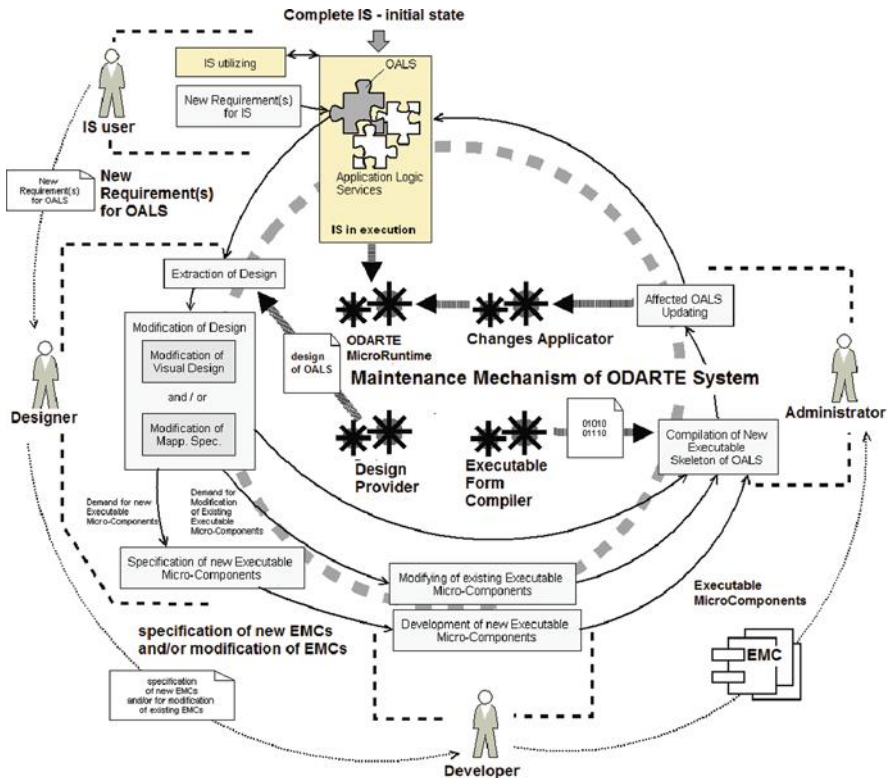


Fig. 4 Life cycle of created ODARTE-compliant IS

is later transformed into XML format which is used by loading new OALS or by modification of existing OALS.

Thus, the OALS designer works only on particular model elements orchestration by designing the visual part of OALS. Then, designer passes the specifications of EMCs which will be needed and which functionality they should contain and provide. Finally, designer writes mapping specification.

Life cycle of prepared ODARTE-compliant IS does not end in “retirement” phase, as it is done by many current IS life cycles – but it continues in enclosed loop representing later development and maintenance (Fig. 4). If the new IS requirement arrived, the next steps are realized after its analysis:

1. OALS design extraction,
2. projection of required changes into visual part of design,
3. adapting of OALS mapping specification in according to changed design of OALS,
4. if it is necessary, requesting new EMCs delivery,
5. recompilation of OALS into runnable, executable form, and
6. OALS re-subsumption into ODARTE-based IS.

Thus, ODARTE-based IS is continually developed, as new application logic requirement arrived from users. At the same time, user of such IS is not totally dependent on creators of the first IS version, because IS design can be whenever extracted and modified by another subject. After modification, the changes appear in behavior of IS. However, there can exist cases in which some new EMCs are required, because existing EMCs are not able to cover required innovations.

The success of modifications depends on an appropriate earlier decomposition of solved problem or appropriate depth of OALS design for later applicable granularity of EMCs (the granularity of EMC must not be too fine-grained or coarse-grained). However, this fact depends on creators of the first IS version and thus ODARTE is not able to affect this problem. This architecture tries to eliminate current problems of IS but not to absolutely solve them (naturally, totally solving of such problems is utopian). The mentioned problem of EMC granularity and OALS decomposition can be eliminated by using of appropriate methodology for workflow or functionality decomposition. In principle, ODARTE has common features with Executable UML (xUML) [11, 13], dynamic software architectures [1], an architectural approach to the automatic composition and adaptation of software components [16], and open source framework SPRING [15].

4 Executable Micro-components and Their Containers

EMCs encapsulate partial functionality of certain kind. This partial functionality is invoked either by code activities (by workflow-based OALS) or by methods calling on wrapper components (by component-based OALS). It inherits from abstract class called EMC. This class is part of micro-framework of runtime environment for ODARTE support and it contains abstract method named `start()` to which code performing required partial functionality will be added. Also, abstract class EMC contains overloaded constructor which takes as its sole parameter hash table containing input parameters for EMC. Analogously, the hash table containing different number and kinds of objects is return value of each implemented EMC. In comparison with threads, we must take into account that each of the EMC functionality invocations (or rather its `start()` method calls) is in current pilot version of ODARTE executed synchronously. This fact is important from EMC using viewpoint. Even if this way of execution is desirable in most of use cases, we do not exclude that asynchronously EMC invocation option will be added in final version of ODARTE.

As we have mentioned, functionality amount which will be added to the EMCs depends on OALS developers because it spools from required activity which OALS will perform, either from OALS composition (that is whether OALS calls another OALSs). Once the visual design of OALS is created together with mapping specification (which completes this visual design by other necessary information) – in the case of workflow-based OALS (Fig. 2b) or mapping specification is created – in the case of component-based OALS (Fig. 2a), the designer specifies required EMCs. Specification of such EMCs is basically the same as a specification of things in

UML models (e.g., in unambiguously textual description form which specifies the inputs/outputs of EMC, its activity). So, the granularity of particular EMCs can be ad hoc very diverse.

Particular EMCs are grouped into so-called container of EMCs (Fig. 5). It has currently the form of dynamic link library on .NET platform. So, it deals with the library which contains at least one class derived from EMC abstract class. They are grouped in such a way that EMCs in one group perform similar activity or contain functionalities with logical relatedness. For example, such container provides functionality for working with some database. In this container one EMC provides connectivity to SQL server, other provides some transaction initializing, yet another provides simple data select from certain table, etc. depending on application logic requirements. Other EMC can contain EMCs for working with XML documents and the other can encapsulate EMCs which provide functionality for securing IS (e.g., application of different encryption mechanisms).

The proceeding of EMCs encapsulation into EMC depends on EMC designers, prospectively again on OALS designer. Let us remark that EMC designers design EMC according to EMC specifications from OALS designer. EMC can also contain other additional classes which are utilized by EMCs – classes derived from EMC abstract class. Final EMC (so all classes, interfaces, and other elements contained in it) is later compiled into .NET assembly form.

Each EMC can be used by multiple OALS. In other words, by one ODARTE-compliant IS we can obtain multiple instances from the same EMC. Also, by mapping specifications we can determine in which OALS certain EMC is used and, so, in case of necessity for EMC changing, we can also determine places in which change of EMC will have an impact.

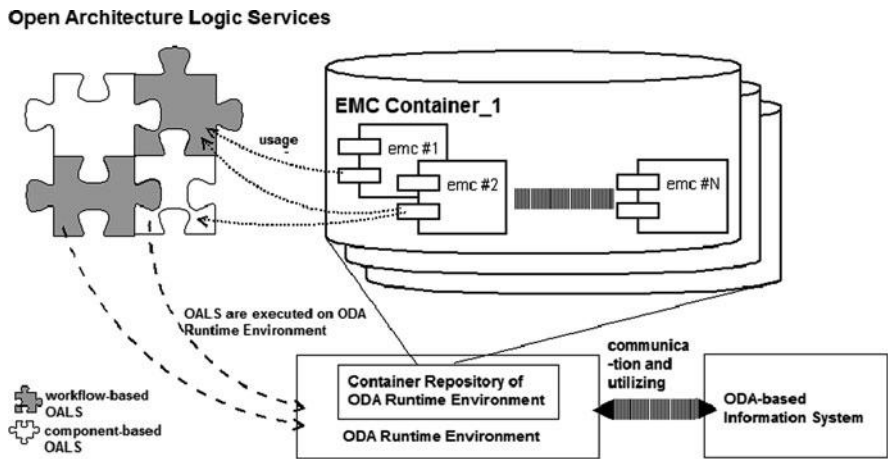


Fig. 5 The container of executable micro-components (EMCs)

5 Runtime Environment

Runtime environment for ODARTE contains special middleware called ODARTE middleware (Fig. 6) which is a mediator between IS and underlying runtime environment (e.g., .NET framework in current pilot version). This middleware consists of these three main parts:

- ODARTE micro-framework,
- ODARTE micro-runtime, and
- ODARTE runtime repository.

ODARTE micro-framework primarily provides possibility of OALS execution by ODARTE middleware. Subcomponent OALS invocator is used for OALS invocation purposes. OALS design extractor provides extraction of particular OALS parts: design (visual and/or mapping specification) and particular EMC extraction. Visual design extractor allows us to obtain visual design of OALS in XOML format (or another format for certain CASE tool with UML support). Mapping specification extractor provides mapping specifications and finally, EMC extractor accesses executable micro-components from their containers. Changes activator is the last subcomponent in ODARTE micro-framework. It is responsible for application of changes in OALS or, in other words, for new/modified OALS loading into IS.

ODARTE micro-runtime is responsible for solely OALS execution and, naturally, for execution of functionality encapsulated in EMCs. It contains micro-runtime controller which manages this part of middleware, so it is responsible for runtime's start, terminating, etc. OALS info provider provides different information concerning OALS in execution (i.e., their actual state of execution, for which user which OALS instances are currently in execution, etc.). OALS executor is responsible for initializing and executing OALS subcomponents.

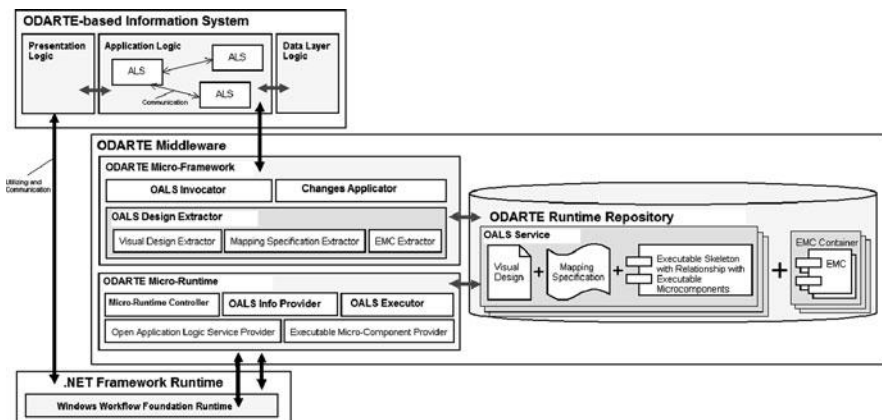


Fig. 6 Runtime environment of ODARTE

ODARTE runtime repository is the last part of ODARTE middleware. It contains solely OALS services, so their particular parts – visual design, mapping specifications, executable skeleton with relationships to EMCs, and other OALSs. The thick black arrows in Fig. 6 describe utilizing and communication between application logic of IS, ODARTE middleware, and underlying runtime environment. The brighter arrows represent communication between internal parts inside IS and middleware. Middleware communicates with its environment by micro-framework. Also, IS utilizes executable runtime environment of .NET framework. Inside IS application logic layer communicates with presentation layer and data layer. ODARTE repository utilizes both micro-framework and micro-runtime. Solely middleware (analogously as IS) utilizes .NET framework runtime [14] as the middleware and is built on it.

6 Experimental Simulation

We prepared the experimental simulation of two components classical ALS and OALS in ODARTE environment [4]. Both tested components run on AMD Athlon 64 3800+/2.4 GHz/2 GB RAM, under Windows Workflow Foundation Beta for Visual Studio 2005, and MS SQL Server 2005. For each sequence of calls (represented on the graph as one point) the same input data are sent directly to both applications, so we skipped the usage of TCP protocol that is to be used in real environment. Tested are two aspects:

- The time of service invocation (standard and ODARTE).
- The time of service execution (standard and ODARTE).

Results are presented on the Fig. 7, where we can see that the time of execution is nearly the same and the invocation time differs slightly for thousands of calls. So slowing down of any application in ODARTE environment is omissible.

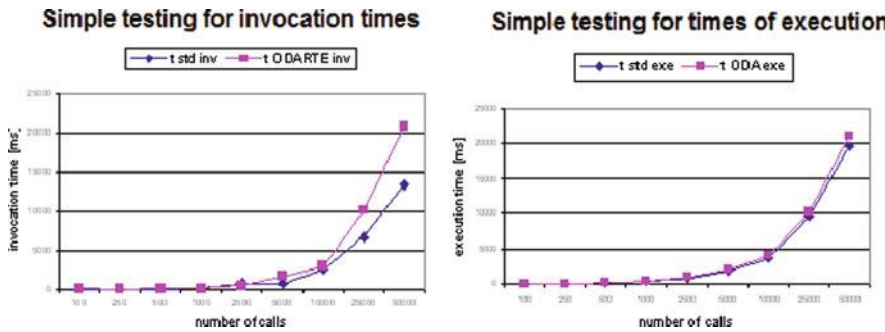


Fig. 7 Results of experimental simulation of classical and ODARTE workflow

7 Conclusions

The chapter presents the ODARTE architecture which is a special type of MDA. We have tried to illustrate that applications can be installed into the ODARTE environment and that can be comparatively easily modifiable. The main difference between classical MDA and ODARTE is that the platform-independent model is strictly connected to the executable part of IS, which stands for platform-specific model. Even if there are virtual machines widely used for application interpretation, here certain quasi-assembler is required. This assembler is then compiled into machine language of target machine by just-in-time compilers. However, here interpretation is not executed only in accordance to source code of virtual machine assembler. For this purpose it is desirable to have certain visual tool at disposal. We want to create it and then embed it into ODARTE management tool in the final version of ODARTE.

Acknowledgement This work has been partially supported by the grants No. 201/09/0990 and No. 201/09/0983 of the Grant Agency of Czech Republic.

References

1. Allen, P., and Frost, S. (1998) *Component-Based Development for Enterprise Systems: Applying the SELECT Perspective (SIGS: Managing Object Technology)*, Cambridge University Press.
2. Arlow, J., and Neustadt, I. (2005) *UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design (2nd Edition)*, Addison-Wesley Professional Press.
3. Atkinson, C., et al. (2005) *Component-Based Software Development for Embedded Systems: An Overview of Current Research Trends (LNCS- 3778)*, Springer, New York, NY.
4. Beličák, M. (2007) Open Design Architecture and Artificial Intelligence Agent Application in Information Systems Practise. *Proc. of the 5th Slovakian-Hungarian Joint Symposium on Applied Machine Intelligence and Informatics*, SAMI 2007, Slovakia, pp. 349–361.
5. Bell, M. (2008) *Service-Oriented Modeling (SOA): Service Analysis, Design, and Architecture*, Wiley, New York, NY.
6. Booch, G., et al. (2005) *The Unified Modeling Language User Guide (2nd Edition)*, Addison-Wesley Press.
7. Ebraert, P., et al. (2005) Pitfalls in unanticipated dynamic software evolution, *2nd ECOOP Workshop on Reflection, AOP and Meta-Data for Software Evolution*, RAM-SE'05, Scotland, pp. 41–49.
8. Frankel, D.S. (2003) *Model Driven Architecture: Applying MDA to Enterprise Computing*, Wiley, New York, NY.
9. Kang, K.C., et al. (1990) *Feature-Oriented Domain Analysis (FODA) Feasibility Study*, Technical Report CMU/SEI-90-TR-21; ESD-90-TR-222, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania.
10. Kleppe, A., et al. (2003) *MDA Explained: The Model Driven Architecture: Practice and Promise*, Addison Wesley Professional Press.
11. Mellor, S., J., and Balcer, M., J. (2002) *Executable UML: A Foundation for Model-Driven Architecture*, Addison Wesley Professional Press.
12. Newcomer, E., and Lomow, G. (2004) *Understanding SOA with Web Services (Independent Technology Guides)*, Addison Wesley Professional Press.

13. Raistrick, C., et al. (2004) *Model Driven Architecture with Executable UML*, Cambridge University Press.
14. De Smet, B.J.F., et al. (2007) Dynamic Workflow Instrumentation for Windows Workflow Foundation. *Proc. of the 2nd International Conference on Software Engineering Advances*, ICSEA'2007, France, pp. 11–16.
15. SPRING Framework Resource Page, URL: <http://springframework.org>.
16. Tivoli, M. (2005) *An architectural approach to the automatic composition and adaptation of software components*, Dissertation, Università di L'Aquila.
17. Unified Modeling Language (UML) Resource Page, URL: <http://www.uml.org/>.

Quality Issues on Model-Driven Web Engineering Methodologies

F. J. Domínguez-Mayo, M.J. Escalona, and M. Mejías

Abstract Nowadays, there are several development methodologies in the field of model-driven web engineering (MDWE) which involve different levels of model-driven architecture (MDA): CIM, PIM, PSM, or code. Attending to the high number of available methodologies, development teams may feel lost when choosing the most suitable one for their projects. Furthermore, proposals usually appear and people feel necessary to evaluate their quality in order to select the appropriate methodology or even to find out the way to improve them. This chapter presents the current work carried out in this field and it is oriented toward the definition of a framework which enables an objective measurement of the proposals' benefits.

Keywords Web engineering · Quality in web engineering · Software metrics · Model-driven development

1 Introduction

Few years ago, several research groups began to analyze the characteristics of new types of software systems that emerged and were known as hypermedia systems, which have evolved to be called web systems. It was the birth of a new line of engineering software that is now called web engineering [8].

Within the paradigm of MDE (model-driven engineering), web engineering is a specific domain in which model-driven software development can be successfully applied [11]. The use of MDE in web engineering is called model-driven web engineering (MDWE), and, as can be noticed through different papers, in the last years several research groups have proposed methodologies with processes, models, and techniques to build applications [9, 14, 23, 26] and it is offering very good results [5, 10, 18, 24].

F.J. Domínguez-Mayo (✉)
University of Seville, Seville, Spain
e-mail: fjdominguez@us.es

Fig. 1 Levels cover by each approach [9]

	CIM	PIM	PSM	Code
HDM			✖	
RMM			✖	✖
OOHDM			✖	✖
WSDM		✖	✖	✖
OO-Method			✖	✖
WebML		✖	✖	✖
UWE	✖	✖	✖	✖
W2000	✖	✖	✖	✖
UWA	✖	✖	✖	✖
OOH		✖	✖	✖
NDT	✖	✖		
OOWS	✖	✖		
WebSA			✖	✖
WebRe	✖			
FLIP/Fusebox 5.1	✖	✖	✖	✖

There are currently several proposals in the literature on MDWE that are very useful for building such applications.

Some of them cover most of the levels and even have tools that support the automation of transformations in the processes of development and evaluation. There are different and varied proposals in web engineering, as shown in Fig. 1 obtained from [9]. In Fig. 1 proposals are shown in rows and levels or phases in columns. A cell without a cross indicates that this approach does not consider this level in its life cycle. A shadowed red cross points out that the proposal includes a phase based on classic proposals, but does not include special proposals for the web. A dark red cross means that the proposal covers the whole level, including specific method and models for the web environment.

This diversity of possibilities and the new trend to use MDE in the proposals open a too wide range of offers and in many cases it may be complex to determine the most appropriate one. Consequently, this chapter presents a first approach to a framework which objectively assesses the proposals for MDWE that a computer has to develop and offers a choice criterion for it.

The chapter is organized into the following sections: Section 2 presents a short introduction about MDWE. Section 3 introduces the problem, motivation, and goals and tries to define a framework that permits quality evaluation of the different methodological proposals. Section 4 provides with the elements to consider in the evaluation of the approaches. Section 5, where the elements have already been identified in the previous section, offers the guidelines to both structure the assessment

and focus on the work plan. Section 6 shows the methodology and work plan determined and a description of the necessary process to achieve the framework. Finally, a set of conclusions and possible future work are established.

2 Model-Driven Web Engineering

Model-driven engineering (MDE) is a software development methodology which consists in the creation of models closer to a particular domain rather than concepts or a specific syntax. The domain environment specific to MDE for web engineering is called model-driven web engineering (MDWE). 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 [19, 20]. MDA was created with the idea of separating the specification of the operational logic of a system from the details that define its uses of the capabilities of the technological platform where it is implemented [19, 20].

Attending to the above mentioned, the goals of MDA are portability, interoperability, and 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 [27].

By applying this paradigm, the life cycle of a software system is completely covered, from requirements capture to its own maintenance, through the generation of the code. MDA distinguishes at least the following stages or levels: CIM, PIM, PSM, and code. This research focuses only on the early stages of development within the CIM and PIM MDWE field. In Fig. 2 a possible model-driven process is applied to web engineering. On the left, the MDE processes are described and on the right models in every level are showed. Orange circles in models represent transformations.

3 Problem, Motivation, and Goals

There are many proposals in the area of MDWE and many comparative studies [22]. Faced with this situation, there is a gap in decision making when an application of a methodology for a real project is required. An important need to assess the quality of existing methodologies therefore arises. On the other hand, the fact of being able to measure these methodologies may facilitate the assessment. The solution of this problem may answer the questions raised above, not only to understand the worth of a proposal but also to have an objective criterion to improve or a possibility of unifying criteria to design new proposals in the future.

The main goal of this research is to lay the basis for defining a framework that allows the quality assessment of different methodological proposals. This work points out the measurement of quality of the proposed MDWE for the first levels

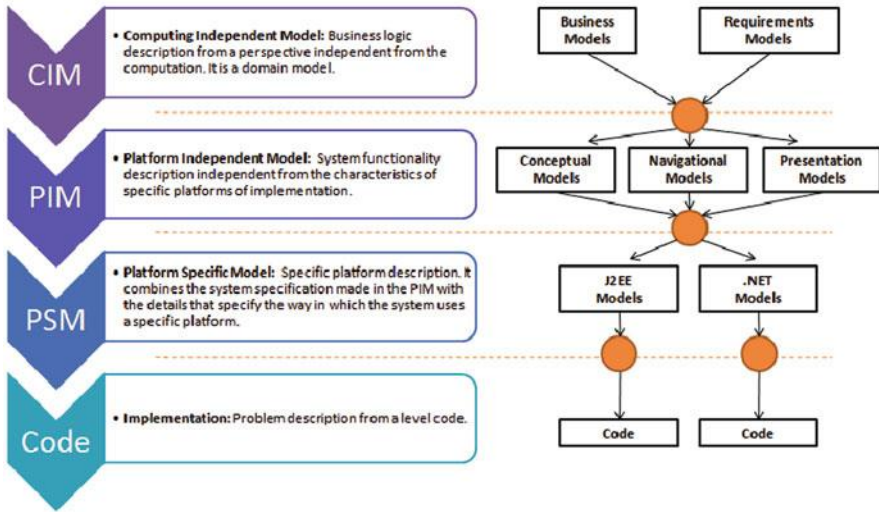


Fig. 2 Model-driven process on web engineering

of the CIM and PIM developments, for the vast majority of existing proposals concentrate on these levels. Thus, our work concentrates on evaluating and comparing existing proposals.

The assessment is based on quantitative or qualitative values which ensure the quality of proposals. As a result, a future objective could be either to unify the criteria to decide on the use of a particular proposal in MDWE or to improve the design of new proposals and the use of standards.

4 Factors to Consider in the Evaluation

It should be highlighted that this research work is in the early stages of development and we still have a very general description of it.

Nevertheless, it will be discussed and identified in general terms the idea of rows and columns kept in mind and considered in future works to define a table or environment necessary to solve the problem defined above. Therefore, for the environment, it would be possible to have a matrix based on the same idea as put forward in [22] which proposes a framework that permits the domain of a distributed information system to be characterized. A matrix has a series of dimensions in its columns and activities from this type of information system in its rows.

The solution of the problem is related to this same idea. The columns, rows, and cells that could become the matrix proposed are described below.

As is visualized in Fig. 3, the columns contain, however, the fundamental aspects of MDWE: metamodels, models (instances of metamodels), and transformations.

				Dimensions			
				Metamodels	Transformations	Models	Total Values
Levels	CIM	Content	Products	Metrics			
		Navigation					
		Presentation					
		Process					
	PIM	Content	Products	Metrics			
		Navigation					
		Presentation					
		Process					
Total Values							

Fig. 3 General idea of the reference environment

For each row every product obtained from the different activities at each level may be analyzed. As it has been mentioned before, the research work will be focused on the study of the CIM and PIM levels. Regarding the study of the CIM level, there are some works which analyze each of the techniques of each activity in detail and they even perform comparative studies among different proposals. At this level, three activities are usually performed: requirements elicitation, requirements specification, and requirements validation [10].

- *Requirements elicitation:* it is the beginning of the process and the stage where all developers gather all necessary information from users and customers. There may be several sources and, as a consequence, products depending on the technique or techniques used are obtained. In [10] the technical highlights are set out: interviews, JAD (joint application development), brainstorming, concept mapping, sketching and storyboarding, use case modeling, and terminology questionnaire and comparison checklist.
- *Requirements specification:* this is the stage where requirements are defined. In the same way as in the earlier stage, products obtained depend on the specific technique. We can see in [10] the techniques used: natural language and ontology glossary, templates, scenarios, use case modeling, formal description, and prototypes.
- *Requirements validation:* at this stage users and customers validate the requirements previously specified. We see again in [10] the techniques used: walk-through or review, audits, traceability matrix for validation, and prototyping.

Products obtained in the CIM level: the use of each of the above techniques dictates whether there are a set of products at the end of each activity, depending on the definition of each proposal and their orientation [10] (process oriented, technology oriented, or product oriented). At this particular level, NDT is a proposal

that stands out since it specifies in detail the technique and product obtained in the requirements. In order to be able to measure, proposals which define products are needed and/or techniques which provide with results. In [10] a detailed study of the results obtained in different proposals is carried out. The products in the CIM level might be classified as content, navigation, presentation, or process as is shown in Fig. 3.

For the study of each PIM level, each proposal (if it is determined in this level) can define a series of products which, in the same way as in CIM, might be generally classified in content, navigation, presentation, or process, as suggested by UWE [14]. The way to classify products keeps on being defined. The products in the PIM level are the result of the application of a transformation from the CIM products.

Products obtained in the PIM level: at this level, it is also possible to classify products as content, navigation, presentation, or process like in CIM. For example, for this level, in UWE there is a metamodel for content and another for navigation, presentation, and for the business process. On the other hand, issues that are widely developed in studies such as those in [24] should be taken into account, and other factors such as the proposal maturity, web modeling, and tools should also be borne in mind. It is relevant to assess all these factors, as they may influence the decision of using a specific proposal.

Finally, the cells should have metrics that indicate either the impact or the influence of each dimension (metamodels, models, and transformations) in the product quality or performance. For example, for metamodels (being interrelated concepts) metrics which measure complexity may be considered. In regards to metrics model, an important study has been revealed in [1]. It proposes a set of metrics for navigational models to analyze the web applications' quality in terms of size and structural complexity. In this chapter, these metrics are defined and validated using a formal framework (DISTANCE) [21] for software measure construction that satisfies the measurement needs of empirical software engineering research. This framework uses the concepts of similarity and dissimilarity among software entities. In DISTANCE, the software attributes are modeled as distances (i.e., conceptual distance) between the software entities that they represent and other ones which act as reference points. These distances are then measured by mathematical functions that satisfy the axiom set of the metric space. DISTANCE could be used to define and theoretically validate all metrics (in metamodel, model, and transformation) in the framework. On the other hand, a general idea about context suitability for CIM and PIM levels might be given in total values when an approach is measured.

5 Structure of the Evaluation

In order to evaluate quality it is necessary to count on instruments that are based on clear definitions. One of these instruments is a quality model (defined in the ISO/IEC 9126). A quality model is defined in ISO as the set of characteristics and the relationships between them which provide the basis for specifying quality requirements

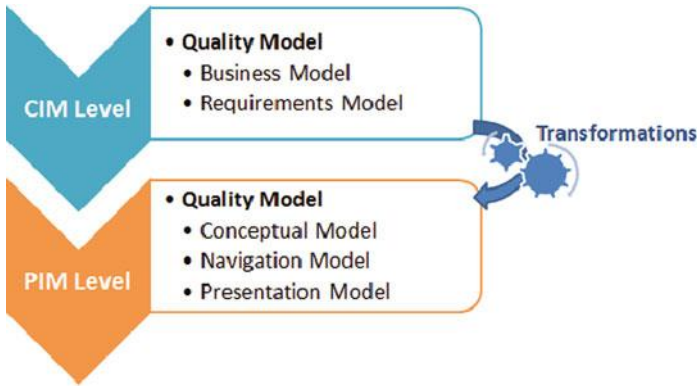


Fig. 4 Quality models which should be kept in mind in every level

and evaluating quality [3]. A different quality model must be designated for each of their products in CIM and PIM levels. In Fig. 4 different models in every level are shown for every product. It may be necessary to define a model quality for every one of them.

As far as the definition of metrics is concerned, in [4] a metrics metamodel is proposed (Fig. 5). A measure may be defined as one base or derived measure; even it may be an indicator which satisfies an information need. On the other hand, a measure is measured using a scale expressed in a unit of measurement. A measure may be defined by an attribute or a lot of them.

Furthermore, the goal/question/metric (GQM) paradigm [2] could be followed for a formal definition of these metrics. In this paradigm, a template is needed to define metrics:

- Analyze – ?
- For the purpose of – ?
- With respect to their – ?
- From the point of view of the – ?
- In the context of – ?

Our aim is to look for series of qualitative and quantitative metrics based on their nature, although it might be interesting to have standard metrics on MDWE which are all, somehow, centralized. In current literature there are many references about metrics [6, 7, 12, 13, 15–17, 25] but till now, nothing has been found to standardize all these.

Another instrument is a quality evaluation process that prescribes how and when quality evaluation must be performed [3]. For the definition of the evaluation process, Fig. 6 shows an assessment process for web engineering adapted from *ISO 14598-1*. As shown in Fig. 6, first, the evaluation requirements are established and, in steps 2 and 3, the evaluation is specified and designed, respectively. Finally, in step 4, the evaluation is executed.

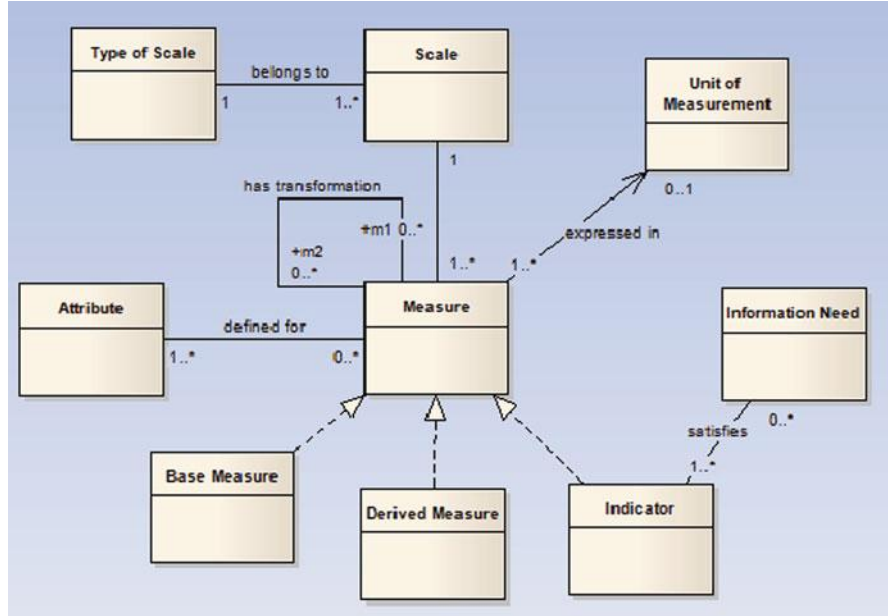


Fig. 5 Metrics metamodel

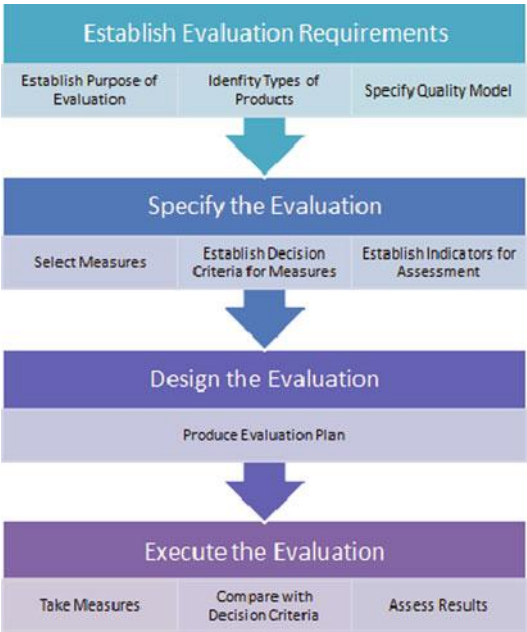


Fig. 6 Evaluation process adapted from ISO 14598 [3]

A controlled evaluation experiment related to this process and existing proposals could be carried out to empirically validate the suggested metrics in the framework.

Again, (GQM) paradigm could be used both to establish evaluation requirements and to specify the evaluation. On the other hand, Spearman correlation coefficient (where the variables can be expressed using an ordinal scale type) may verify the dependency among variables (metrics proposed to measure products and results like quality characteristics such as usability, accessibility, maintainability)

On the other hand, for the evaluation of the proposals, the scheme revealed in [3] which introduces a complete process (CIM, PIM, PSM, and code) to measure quality in web engineering must be applied.

6 Framework Definition Process and Conclusions

A scientific and technical research methodology is followed. The work plan includes conducting a state-of-the-art study on current topics of research:

- Metamodels, models (for instance the meta), and transformations
- Quality in metamodels, models, and transformations
- Consideration of project proposals MDWE, tools, etc.
- Comparative studies of proposals MDWE
- Frameworks designed to measure quality
- Quality in software engineering (development, models, meta, change, etc.)
- Study and definition of metrics and indicators for meta models and their transformations and metric engineering ontologies on web
- Processes for assessing quality in web engineering

The outline of the work plan corresponds to that shown in Fig. 7. First, an environment to measure the value of proposals and the definition of a process to evaluate CIM and PIM levels should be specified. In steps 2 and 3 MDWE proposals should be compared in an iterative way to obtain conclusions on the evaluations. An iterative feedback process (proposals – measurement and evaluation – conclusions) should improve as much as possible the work environment.

Bearing in mind that there are major work to be carried out and that this research is still in the early stages of development, we trust that good results will permit future research to improve the value of existing proposals in MDWE. Furthermore, the use of standards seems to be essential for the research development in this type of problem.

A framework which permits us to measure quality or adaptability for an approach or methodology given a context might be useful because it can help development teams to choose the most suitable one for every project. They would have an environment to decide which proposal is the most appropriated.

To define metrics, (GQM) paradigm might be followed for a formal definition, and these metrics might be defined and validated using the formal framework

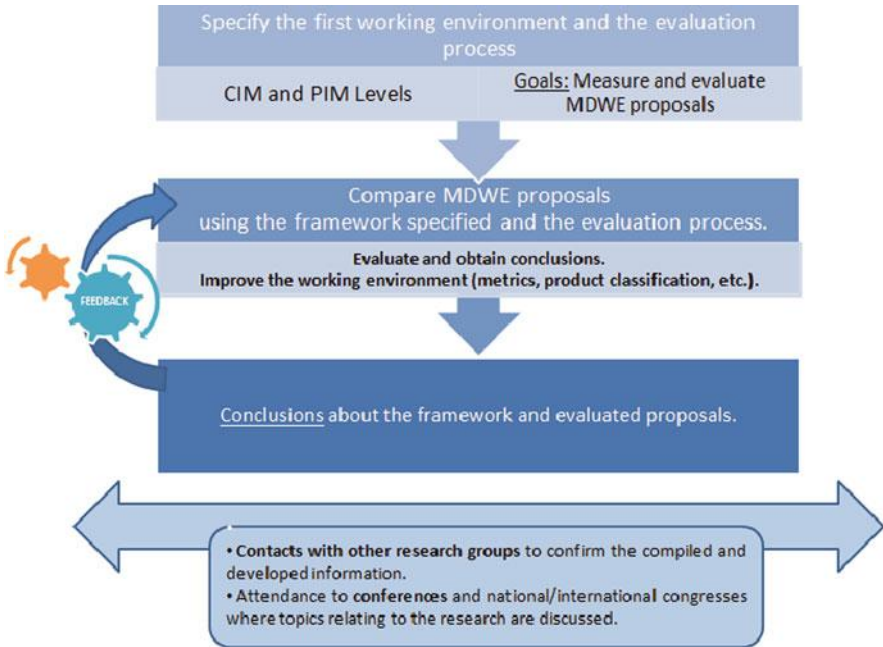


Fig. 7 General work plan

(DISTANCE) [21]. On the other hand, quality models (for products in CIM and PIM levels in every dimension) and an evaluation model (for the complete assessment) should be defined to ensure good results.

In regards to the contributions obtained from this research, a generic environment is required for the measurement of the value of MDWE proposal in order to be able to assess and improve their quality or adaptability. In this way, criteria can be unified when developing a new methodology or improving current proposals.

Acknowledgments This research has been supported by the project QSimTest (TIN2007-67843-C06_03) and by the RePRIS project of the Ministerio de Educación y Ciencia (TIN2007-30391-E), Spain.

References

1. S. Abrahão, N. Condori-Fernández, L. Olsina and O. Pastor (2003) Defining and Validating Metrics for Navigational Models. IEEE Computer Society. *Proceedings of the Ninth International Software Metrics Symposium (METRICS'03)*. pp. 200–210, ISSN: 1530-1435, ISBN: 0-7695-1987-3.
2. V. Basili and H. Rombach (1988) The TAME Project: towards improvement-oriented software environments. *IEEE Transactions on Software Engineering* 14, pp. 758–773.
3. C. Cachero, G. Poels and C. Calero (2007) Towards a quality-aware web engineering process. *Twelfth International Workshop on Exploring Modelling Methods in Systems Analysis and Design*, 1, pp. 7–16. Held in conjunction with CAISE'07 Trondheim.

4. C. Cachero, G. Poels, C. Calero and Y. Marhuenda (2007) Towards a quality-aware engineering process for the development of web applications. Working Papers of Faculty of Economics and Business Administration, Ghent University, Belgium 07/462, Ghent University, Faculty of Economics and Business Administration.
5. C. Calero, J. Ruiz and M. Piattini (2004) *A Web Metrics Survey Using WQM*. ICWE 2004, LNCS 3140, pp. 147–160.
6. S.R. Chidamber and C.F. Kemerer (1991) Towards a metrics suite for object oriented design, in A. Paepcke, (ed.) *Proc. Conference on Object-Oriented Programming: Systems, Languages and Applications (OOPSLA 91)*. ACM, New York, NY, USA. Vol. 26, Issue 11, pp. 197–211, ISSN:0362-1340.
7. S.R. Chidamber and C.F. Kemerer. (1994) *A Metrics Suite for Object Oriented Design*, *IEEE Transactions on Software Engineering*. IEEE Press Piscataway, NJ, USA, Vol. 20, Issue 6, pp. 476–493, ISSN:0098-5589.
8. Y. Deshpande, S. Marugesan, A. Ginige, S. Hanse, D. Schawabe, M. Gaedke and B. White (2002) Web Engineering, *Journal of Web Engineering*, 1(1), pp. 3–17.
9. M.J. Escalona and G. Aragón (2008) *NDT. A model-driven approach for web requirements*. *IEEE Transactions on Software Engineering*, San Francisco, CA, USA, pp. 377–390, ISSN: 0098-5589.
10. M.J. Escalona and N. Koch. (2004) Requirements engineering for web applications – a comparative study. *Journal of Web Engineering*, 2(3), pp. 193–212.
11. J. Fons, V. Pelechano, M. Albert and O. Pastor (2003) Development of web applications from web enhanced conceptual schemas, *Proceedings of the 22nd International Conference on Conceptual Modeling*. I.-Y. Song et al. (Eds.): ER 2003, LNCS 2813, pp. 232–245.
12. F. García, M. F. Bertoa, C. Calero, A. Vallecillo, F. Ruíz, M. Piattini and M. Genero (2005) Towards a consistent terminology for software measurement. *Information and Software Technology*, 48, pp. 631–644.
13. B. Henderson-Sellers (1996) *Software Metrics*, Prentice Hall, Hemel Hempstead, UK.
14. C. Kroiß and N. Koch (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.
15. A. Lake and C. Cook (1994) Use of factor analysis to develop OOP software complexity metrics. *Proceedings of The 6th Annual Oregon Workshop on Software Metrics*, Silver Falls, Oregon.
16. Y.-S. Lee, B.-S. Liang, S.-F. Wu and F.-J. Wang (1995) Measuring the coupling and cohesion of an object-oriented program based on information flow, *Proc. International Conference on Software Quality*, Maribor, Slovenia.
17. M. Lorenz and J. Kidd (1994) *Object-Oriented Software Metrics*, Prentice Hall Object-Oriented Series, Englewood Cliffs, NJ.
18. N. Moreno, P. Fraternali and A. Vallecillo (2006) *A UML 2.0 Profile for WebML Modeling*, ICWE'06 Workshops.
19. OMG: MDA Guide (2005) <http://www.omg.org/docs/omg/03-06-01.pdf>
20. J. M. Pérez, F. Ruiz, M. Piattini. (2007) *Model Driven Engineering Aplicado aBusiness Process Management*, Informe Técnico UCLM-TSI-002.
21. G. Poels and G. Dedene. (1999) *DISTANCE: A Framework for Software Measure Construction*. Research Report 9937, Department of Applied Economics, Catholic University of Leuven
22. J. Ralyté, X. Lamielle, N. Arni-Bloch and M. Léonard (2008) Distributed Information Systems development: A Framework for Understanding and Managing. *International Journal of Computer Science and Applications, Technomathematics Research Foundation*, 5(3b), pp. 1–24.
23. A. Schauerhuber, M. Wimmer and E. Kapsammer (2006) Bridging existing web modelling languages to model-driven engineering: a metamodel for WebML. *International Conference On Web Engineering; Vol. 155. Workshop proceedings of the sixth international conference on Web engineering*. Palo Alto, CA (MDWE'06). ISBN:1-59593-435-9.

24. W. Schwinger, W. Retschitzegger, A. Schauerhuber, G. Kappel, M. Wimmer, B. Pröll, C. Cachero Castro, S. Casteleyn, O. De Troyer, P. Fraternali, I. Garrigos, F. Garzotto, A. Ginige, G-J. Houben, N. Koch, N. Moreno, O. Pastor, P. Paolini, V. Pelechano Ferragud, G. Rossi, D. Schwabe, M. Tisi, A. Vallecillo, van der Sluijs and G. Zhang. (2008) A survey on web modeling approaches for ubiquitous web applications. *International Journal of web Information Systems*, 4(3), pp. 234–305.
25. Sdmetrics, <http://www.sdmetrics.com/>
26. A. Vallecillo, N. Koch, C. Cachero, S. Comai, P. Fraternali, I. Garrigós, J. Gómez, G. Kappel, A. Knapp, M. Matera, S. Meliá, N. Moreno, B. Pröll, T. Reiter, W. Retschitzegger, J. E. Rivera1, A. Schauerhuber, W. Schwinger, M. Wimmer and G. Zhang (2007) MDWEnet: A Practical Approach to Achieving Interoperability of Model-Driven Web Engineering Methods. :“7th International Conference on Web Engineering, Workshop Proceedings”, Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy, pp. 246–254, ISBN: 978-88-902405-2-2.
27. Wikipedia, http://en.wikipedia.org/wiki/Model-driven_engineering (May 2009).

Measuring the Quality of Model-Driven Projects with NDT-Quality

M.J. Escalona, J.J. Gutiérrez, M. Pérez-Pérez, A. Molina, E. Martínez-Force, and F.J. Domínguez-Mayo

Abstract Model-driven web engineering (MDWE) is a new paradigm which provides satisfactory results in the development of web software systems. However, as can be concluded from several research works, MDWE provokes traceability problems and the necessity of managing constraints in metamodel instances and transformation executions. The management of these aspects is usually executed manually in the most of MDWE approaches. Nevertheless, model-driven paradigm itself can offer suitable ways to manage them. This chapter presents NDT-Quality, an approach to measure the quality of web projects developed with NDT (navigational development techniques), and offers a view about the application of this tool in real web projects.

Keywords Web engineering · Model driven web engineering · Quality assurance · Tool support

1 Introduction

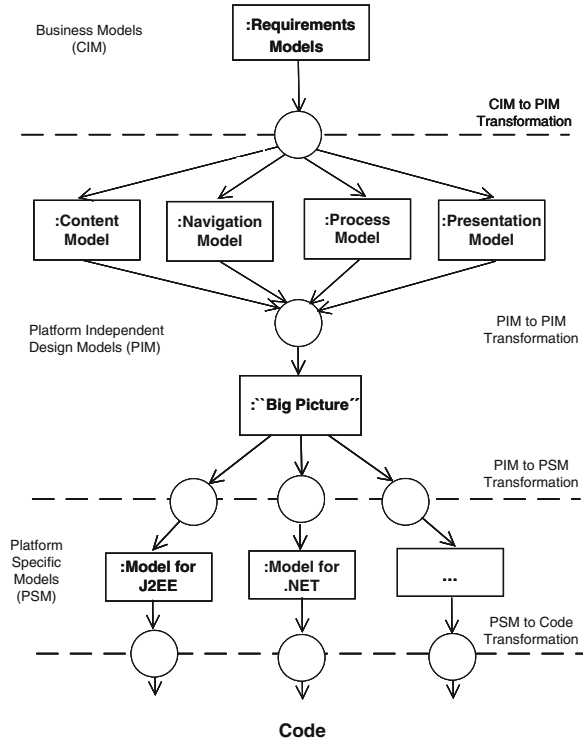
Model-driven engineering is a new paradigm that is being assumed by several research groups in the improvement of methodological approaches for web environment. UWE (UML web engineering) [12], WebML (web modeling languages) [2], and OOH [1] are only a few examples. MDWE is providing good results in this area, but for application to real projects some tools are necessary to assure the quality of results.

MDWE consists in the definition of a set of metamodels in each phase of the life cycle, followed by the establishment of a set of transformations between these metamodels which enable subsequent models to be derived.

For instance, Fig. 1, obtained from [14], shows a schema on the adaptation of standard MDA (model-driven architecture) [15] in web development.

M. Escalona (✉)
University of Seville, Seville, Spain
e-mail: mjescalona@us.es

Fig. 1 Model-driven web engineering



In this environment, a set of metamodels on CIM (computer-independent model) level, requirements models, permit the capture of requirements information.

With CIM-to-PIM transformations, analysis models can be systematically obtained: content model, navigational model, etc. On the PIM (platform-independent model) level, some new transformations (PIM-to-PIM) can be applied in order to obtain design models.

Subsequently, in PSM (platform-specific model) models can also be obtained from the PIM level.

Finally, a code could be generated from PSM models with PSM-to-code transformations.

MDWE shows important advantages in software development. Transformations assure traceability among levels. Furthermore, the systematic generation of models based on early models can reduce the development time and, if suitable tools are defined, this process could even be automatic.

However, some questions arise. One of the most important is: What happened to the changes? For instance, if the analyst detects a problem and changes the model after the content model generation, then the traceability of requirements could be lost. To execute CIM-to-PIM, transformations may be considered again, but even a little change in some part may imply some delay or extra work.

The aim of this chapter is oriented toward solving these maintenance problems in MDWE. It presents a MDWE methodology, NDT (navigational development

techniques) [11], in Section 2. Section 3 introduces one of its associate tools, NDT-Quality. NDT-Quality manages the maintenance of traceability and the assurance of quality in model-driven applications. A real application in a public organization in Spain is given. Related work and conclusions are drawn in the final section.

2 NDT – Navigational Development Techniques

NDT [9] is a methodological web process focused on both the requirement and the analysis phases. NDT offers a systematic way to deal with the special characteristics of the web environment. NDT is based on the definition of formal metamodels that allow derivation relations to be created between models. NDT takes this theoretic base and enriches it the elements necessary for the definition of a methodology: techniques, models, methodological process, etc. in order to offer a suitable context for its application to real projects.

Figure 2 presents the life cycle of NDT. Initially, NDT only covered the requirement and the analysis phases.

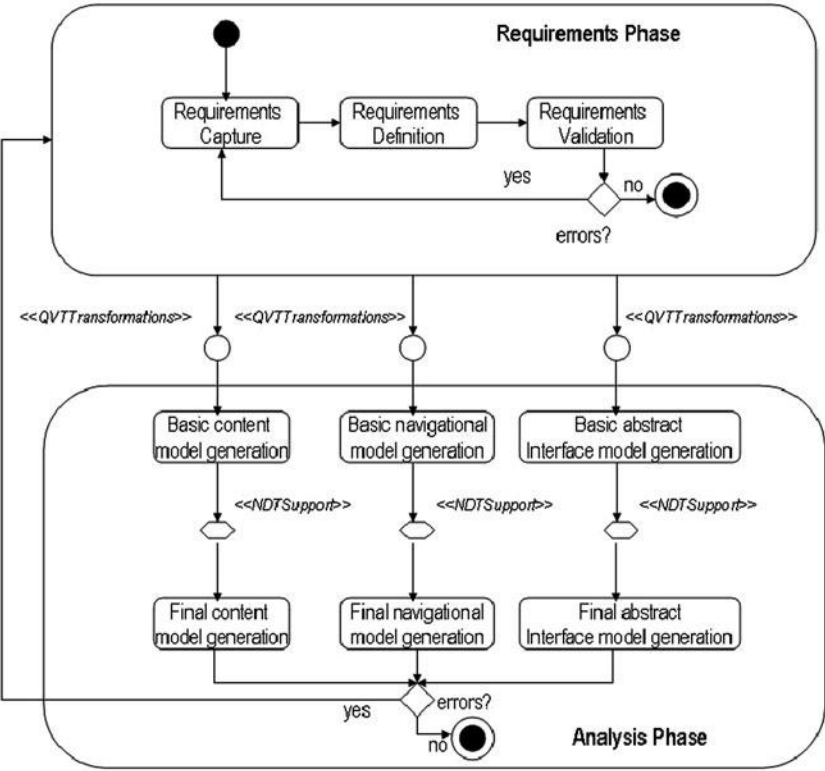


Fig. 2 Life cycle of NDT

The capture, definition, and validation of requirements are involved into the requirements phase. To this end, NDT proposes the division of requirements into different groups depending on their nature: storage information requirements, functional requirements, actors' requirements, interaction requirements, and non-functional requirements. To cope with each kind of requirement, NDT proposes the use of special patterns [4] and UML [18] techniques, such as the use of case techniques. Requirements in NDT are formally presented in a requirement metamodel where some constraints and relations are defined.

The life cycle then passes to the analysis phase. NDT proposes three models in this phase: the conceptual model, the navigational model, and the abstract interface model. The conceptual model of NDT is represented in the methodology using the class diagram of UML and the other two models are represented by UWE notation [13].

The class diagram of UML and the navigational and the abstract interface of UWE have their own metamodels. From the requirement metamodels and analysis metamodels, NDT defines a set of QVT transformations that are represented in the figure with the QVTTransformation stereotype. Thus, the shift from requirements to analysis in NDT is a systematic method based on these formal transformations. The direct application of these transformations generates a set of analysis models known in NDT as the basic analysis models. After the systematic generation, the analyst group can change these basic models by adding new relations, attributes, etc. that improve the models. This step depends on the analyst's knowledge and is presented in the figure with the stereotype NDTSupport. This improvement generates the final analysis models. This second step is not systematic. However, NDT has to ensure that agreement between requirement and analysis models is maintained. Hence, this step is controlled by a set of rules and heuristics defined in NDT.

After the analysis model has been created, the development process can continue with another methodology, such as UWE or OOHDM (object-oriented hypermedia design method) [19], in order to obtain the code.

NDT offers a suitable environment for the development of web systems. It offers specific techniques to deal with critical aspects in the web environment. If a correlation with MDA (see Fig. 1) is made, NDT presents a CIM in the requirements phase; a set of PIMs in the analysis phase; and a set of formal transformations among them.

NDT has been widely applied in practical environments and has achieved very good results since it reduces the development time with the application of transformations and ensures agreement between requirements and analysis. In [7] the practical evolution of NDT is presented together with some of the most important practical applications.

3 NDT-Quality

The application of MDWE and transformations is difficult and quite expensive if it lacks in a set of tools which automate the procedure.

NDT-Suite¹ consists in a set of tools defined to support the development of web systems with NDT and it is composed of four tools:

1. *NDT-Profile*: This is a specific profile for NDT, developed using Enterprise Architect [10]. This tool offers an environment to define specific profiles, and NDT-Profile has adapted Enterprise Architect to support each artifact of NDT.
2. *NDT-Driver*: This is a tool to execute transformations of NDT. NDT-Driver is a Java-free tool which implements QVTTransformations (see Fig. 2) and allows analysis models to be obtained automatically from the requirements model.
3. *NDT-Quality*: This is a tool that checks the quality of a project developed with NDT-Profile.
4. *NDT-Report*: This is a tool that prepares formal documents in order to be validated by final users and clients. For instance, it enables the automatic generation of a requirements document with the format defined by clients.

3.1 The Necessity of NDT-Quality

One of the most relevant characteristics of NDT and NDT-Suite is their practical application. The real application of MDE provides an important source of knowledge for the improvement and adaptation of the methodology and its associate tools.

NDT-Quality was developed as a practical necessity when NDT-Profile started to be applied in real companies.

Although NDT-Profile offers a suitable environment to NDT and it manages the use of artifacts and UML and NDT constraints, development teams could still make errors and inconsistencies in the definitions of systems.

When a development team develops a system with NDT-Profile, they create requirements, classes, use cases, etc. with the tool box defined in Enterprise Architect to deal with NDT.

This product must be checked in order to assure two important parts:

1. The quality of the use of NDT in each development phase.
2. The quality of the traceability with the MDE rules of NDT.

In the first group, NDT-Quality checks the use of any artifact of the methodology. For instance, NDT says that each storage information requirements has to be named and described. NDT-Quality checks that every storage information requirement has a name and a description.

When a development team finishes the requirements phase and generates the analysis phase with NDT-Driver, they can make changes to adapt the results. Some of these changes are allowed in NDT, even though the traceability must be assured.

¹All these tools and their manuals can be downloaded from www.iwt2.org. This chapter focuses on the presentation of NDT-Quality.

Therefore, NDT-Quality also checks this traceability. It compiles a set of rules to ensure that NDTSupport rules (see Fig. 2) are kept in final models.

In the enterprise environment a tool such as NDT-Quality is crucial. The execution of MDE transformations is only possible if metamodels are correctly implemented whereby the rules of the methodology are followed. NDT-Quality guarantees the quality in the application of NDT.

3.2 The Interface of NDT-Quality

NDT-Quality is completely based on NDT-Profile. A working group who wants this environment to be used has to define the project through NDT-Profile and then saves it in an Enterprise Architect file.

The interface of NDT-Quality is quite simple. Figure 3 shows the main interface of the tools. NDT-Quality, like all other NDT-Suite tools, is available in both Spanish and English.

In Project Name section, the name of the project must be introduced. With the search button, the Enterprise Architect file with the project must be selected and, with the set of checks on the left of the screen, the user can select which part of the project must be checked by NDT-Quality.

Although NDT only focused on the first phases of the life cycle, requirements, and analysis, a practical solution of the tool was prepared to work in the enterprise environment. It was presented in [8]. This extension of NDT covers all phases of the life cycle. For this reason, in the screen, NDT-Quality now provides the user with several options:

1. *Requirements, analysis, design, and tests*: If any of these checks are selected, NDT-Quality checks artifacts and NDT rules in this phase. For instance,

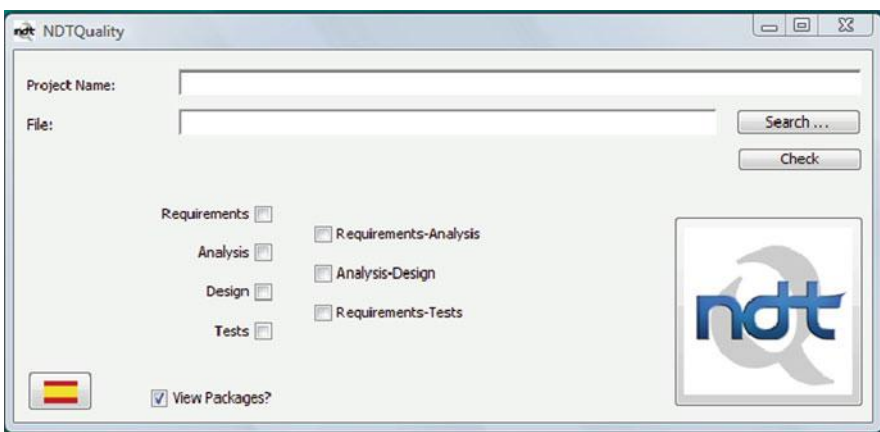


Fig. 3 NDT-Quality main interface

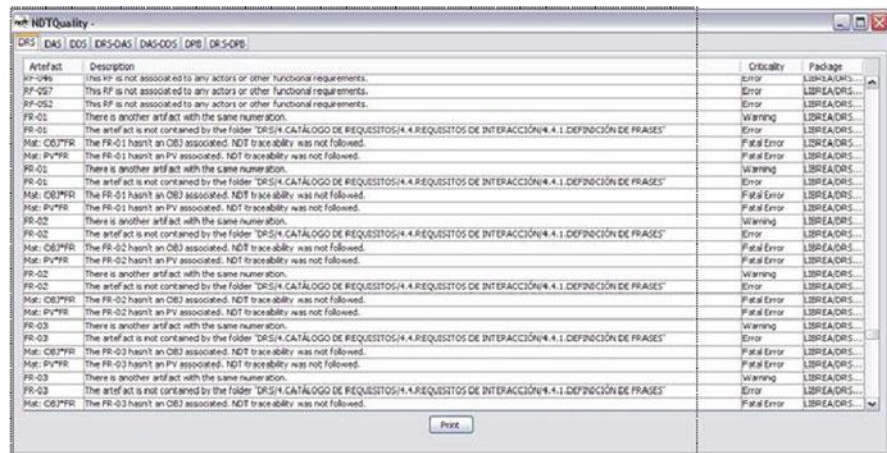
- if we want to check the requirements phase, we must select “requirements check.”
2. *Requirements analysis, analysis design, and requirements test:* These options enable the MDE traceability between phases to be checked. For instance, if we select requirements analysis, NDT-Quality checks the rules to assure the traceability between both the analysis metamodels and requirements metamodels.
 3. *View package:* It is an option to see the package where an artifact with names is defined. When these options are selected, and with the check bottom pressed, NDT-Quality starts the checking process.

The response time in NDT-Quality is quite short. As is presented in Section 2.3, it evaluates the NDT rules on a database. For a project with about 100 requirements it takes only a few seconds.

The checked results are presented on a screen similar to that shown in Fig. 4. For each check selected NDT-Quality presents tabs with the check results. For instance, in Fig. 4, the mistakes found in the requirements phase are presented in the first tab.

The columns of this report are the following ones:

- Artifact:* The artifact where the error was found.
- Description:* A description of the problem.
- Criticality:* An indication of the grade of importance of the mistake. They can be warning, error, or fatal error. With errors or fatal errors NDT-Driver cannot be used.
- Package:* This option is presented only if the check in Fig. 3 is selected. It shows the path in the NDT-Profile file where the artifact is located.



Artifact	Description	Criticality	Package
RF-046	This RF is not associated to any actors or other functional requirements.	Error	LIBREAQRS...
RF-057	This RF is not associated to any actors or other functional requirements.	Error	LIBREAQRS...
RF-052	This RF is not associated to any actors or other functional requirements.	Error	LIBREAQRS...
FR-01	There is another artifact with the same numeration.	Warning	LIBREAQRS...
FR-01	The artifact is not contained by the folder "DRS/H.CATALOGO DE REQUISITOS/H.4.REQUISITOS DE INTERACCIÓN/H.4.1.DEFINICIÓN DE FRASES"	Error	LIBREAQRS...
Mat: OB/PR	The FR-01 hasn't an OB associated. NDT traceability was not followed.	Fatal Error	LIBREAQRS...
Mat: PV/PR	The FR-01 hasn't an PV associated. NDT traceability was not followed.	Fatal Error	LIBREAQRS...
FR-02	There is another artifact with the same numeration.	Warning	LIBREAQRS...
FR-02	The artifact is not contained by the folder "DRS/H.CATALOGO DE REQUISITOS/H.4.REQUISITOS DE INTERACCIÓN/H.4.1.DEFINICIÓN DE FRASES"	Error	LIBREAQRS...
Mat: OB/PR	The FR-02 hasn't an OB associated. NDT traceability was not followed.	Fatal Error	LIBREAQRS...
Mat: PV/PR	The FR-02 hasn't an PV associated. NDT traceability was not followed.	Fatal Error	LIBREAQRS...
FR-02	There is another artifact with the same numeration.	Warning	LIBREAQRS...
FR-02	The artifact is not contained by the folder "DRS/H.CATALOGO DE REQUISITOS/H.4.REQUISITOS DE INTERACCIÓN/H.4.1.DEFINICIÓN DE FRASES"	Error	LIBREAQRS...
Mat: OB/PR	The FR-02 hasn't an OB associated. NDT traceability was not followed.	Fatal Error	LIBREAQRS...
Mat: PV/PR	The FR-02 hasn't an PV associated. NDT traceability was not followed.	Fatal Error	LIBREAQRS...
FR-03	There is another artifact with the same numeration.	Warning	LIBREAQRS...
FR-03	The artifact is not contained by the folder "DRS/H.CATALOGO DE REQUISITOS/H.4.REQUISITOS DE INTERACCIÓN/H.4.1.DEFINICIÓN DE FRASES"	Error	LIBREAQRS...
Mat: OB/PR	The FR-03 hasn't an OB associated. NDT traceability was not followed.	Fatal Error	LIBREAQRS...
Mat: PV/PR	The FR-03 hasn't an PV associated. NDT traceability was not followed.	Fatal Error	LIBREAQRS...
FR-03	There is another artifact with the same numeration.	Warning	LIBREAQRS...
FR-03	The artifact is not contained by the folder "DRS/H.CATALOGO DE REQUISITOS/H.4.REQUISITOS DE INTERACCIÓN/H.4.1.DEFINICIÓN DE FRASES"	Error	LIBREAQRS...
Mat: OB/PR	The FR-03 hasn't an OB associated. NDT traceability was not followed.	Fatal Error	LIBREAQRS...

Fig. 4 NDT-Quality results screen

3.3 The Architecture of NDT-Quality

Enterprise Architect is supported by a database (Access, Oracle, or MySQL can be configured) with a specific relational structure.

When a development team uses NDT-Profile, each element defined in the project is stored in the Enterprise database.

NDT-Quality was developed using Java. NDT-Quality has a specific set of rules that must be applied in each phase and are relevant in the project.

3.4 Practical References

NDT-Suite has been applied to several real projects. In fact, nowadays, some public organizations like Culture Government in Andalusia [3] or Emasesa [5] are working with this tool environment to develop their web projects.

One of the first projects where NDT-Quality started to be used was the project for the digitalization and diffusion of the historical archive of the Medinaceli Duke Foundation. This project was developed by a company for the Culture Andalusia Government and it was aimed at spreading documental resources of this famous family via Internet.

This historical archive consists in a set of documental resources produced and received from any noble house, states, or historical patrimony that were included in Medinaceli family through marriage or family alliances.

This application contains three important modules:

1. *A module to manage any document:* It allows the introduction of new documents into the systems, their description, and location in the system. This functionality can only be executed by the administrator.
2. *A module to manage any search in the system:* With a historical researcher, a tourist, or anybody interested in this historical archive, who wants to search for something in the system, this module offers an efficient and a very intuitive way to manage any kind of search.
3. *A module to download documents from the system:* The system provides copies of documents for research and work purposes. Some special users can use this function and obtain information to include in their studies.

The first phase completed in the project was the requirements phase. In the first presentation of the NDT-Profile file, the company runs without NDT-Quality. The number of errors appears in the first row of Table 1. NDT-Quality was presented to the company in the representation of the requirements phase, the number of error decreased to only one warning. This implementation was quite simple and the requirements phase was accepted.

The company then used NDT-Driver in order to generate the analysis phase. In the first presentation of the analysis phase only three errors were detected.

Table 1 Number of errors detected in each phase

Phase	Warning	Errors	Fatal errors	Comments
Requirements v. 1.00	13	6	3	Without NDT-Quality
Requirements v. 2.00	1	0	0	With NDT-Quality
Analysis v. 1.00	0	3	0	MDE traceability problems
Analysis v. 2.00	0	0	0	

NDT-Quality was used and they solved their problems before presenting the results. These three errors (third row in the table) were produced by MDE traceability problems. Once it was explained that the relations must be kept, they solved the problem easily. In the design phase the number of errors was 0.

As can be concluded from the table, the number of errors was reduced with the use of NDT-Quality. This tool is highly suitable for measuring quality by the final client, in this example the Culture Government, or by their own software provider that without any additional cost can offer a more suitable result.

In other projects where NDT-Quality was used, the results were similar. The improvement in the results was very important since both providers and final users have the same objective and an automatic way to measure the quality of the project.

4 Related Works

Very few related work can be found on the web or in literature. In fact, although MDWE is being assumed by several web engineering methodologies, the use of tools to measure the quality of its application is scarce.

For instance, UWE methodology provides a set of metamodels for each of its phases [13]. It offers a set of transformations and a tool, named MagicUWE [6], oriented to support the MDE development with UWE. However, no special support to manage the quality of models developed is included.

Similarly for WebML, several metamodels were developed regarding this methodology [16, 20], and its transformations, mainly oriented toward code generation, offer a suitable and very powerful MDE environment. Nevertheless, the measure of quality was something last.

In fact, in the last MDWE workshop [17], the issue of measuring the quality when MDWE is applied was established as a high-priority procedure.

In the literature, several approaches to measure the quality of software models can be found, for instance, SDMetrics [21], an environment with a set of metrics to measure the quality of each UML model. In fact, none of them offers special metrics for the MDE paradigm.

5 Conclusions

This chapter presents NDT-Quality, a tool to measure the quality of the application of a MDE methodology, NDT.

The chapter presents the problem in Section 1 and then introduces NDT and details of NDT-Quality. It also describes a global view of how NDT-Quality reduces project mistakes.

The scarcity of research in this area is pointed out. There are several MDWE methodologies and standard frameworks with guides to measure the quality of the system. However, the use of metrics and methods of measuring the quality of MDE applications needs further study.

As future work, the evolution of NDT and NDT-Suite is our highest priority. With the continuous feedback obtained from our practical applications, these tools are frequently being improved.

For instance, one important aspect is to adapt our methodological environment to technological evolution projects. That is, our environment is oriented toward requirements and continues with the classic life cycle. However, what happens when an old application, without documentation or models, needs to be migrated into new technology? or what happens to the maintenance of a project? and how can knowledge be reused and kept if only a part is to be changed?

These questions open new research lines to better our tools and our methodology.

Acknowledgments This research has been supported by the project QSimTest (TIN2007-67843-C06_03) and by the RePRIS project of the Ministerio de Educación y Ciencia (TIN2007-30391-E), Spain.

References

1. C. Cachero. Una extensión a los métodos OO para el modelado y generación automática de interfaces hipermediales. PhD Thesis. University of Alicante. Alicante, Spain 2003.
2. S. Ceri, P. Fraternali and P. Bongio. Web Modelling Language (WebML): a modelling language for designing web sites. *Conference WWW9/Computer Networks*. 33(1–6), pp. 137–157. Mayo 2000.
3. Consejería de Cultura. Junta de Andalucía. www.juntadeandalucia.es/ccul.
4. Durán, B. Bernárdez, A. Ruiz and M. Toro. A requirements elicitation approach based in templates and patterns. *Workshop de Engenharia de Requisitos*. Buenos Aires, Argentina. 1999.
5. Emasesa. Empresa Municipal de Aguas de Sevilla. <http://www.aguasdesevilla.com>.
6. MagicUWE. <http://www.pst.informatik.uni-muenchen.de/projekte/uwe/toolMagicUWE.html>.
7. M.J. Escalona, J.J. Gutierrez, D. Villadiego, A. León and A.H. Torres. Practical experience in web engineering. *Advances in Information System Development*. New Methods and Practice for the Networked Society. ISBN: 13-978-0-387-70801-0. V.2. pp. 421–434. 2007.
8. M.J. Escalona, J.J. Gutiérrez, J.A. Ortega, and I. Ramos. HYPERLINK “publicaciones.php?id=81” NDT & METRICA V3. An approach for public organizations based on Model Driven Engineering WEBIST 2008 Proceedings of the 4th International conference on web information systems; Portugal (2008), Vol. 1, pp. 224–227, ISBN: 978-989-8111-26-5
9. M.J. Escalona, M. Mejías, J. Torres, and A.M. Reina. HYPERLINK “publicaciones.php?id=140” The NDT development process, Lecture Notes in Computer Science. Springer Verlag; Alemania (2003), Vol. 2722, pp. 463–467, ISBN: 0302-9743, 2003
10. Enterprise Architect. www.sparxsystems.com
11. M.J. Escalona and G. Aragón. NDT A Model-Driven approach for Web requirements. *IEEE Transaction on Software Engineering*. 34(3). pp. 370–390. 2008.

12. N. Koch. Software engineering for adaptive hypermedia applications. Ph. Thesis, FAST Reihe Softwaretechnik, Vol. 12, Uni-Druck Publishing Company, Munich. Germany. 2001.
13. N. Koch. Transformation techniques in the model-driven development process of UWE. International Conference On Web Engineering; Vol. 155. Workshop proceedings of the sixth international conference on Web engineering. Palo Alto, California. WORKSHOP SESSION: *Second international workshop on model driven web engineering (MDWE'06)*, Article No.: 3. 2006 ISBN:1-59593-435-9, 2006.
14. N. Koch, G. Zhang and M.J. Escalona. Model transformations from requirements to web system design. *ACM International Conference Proceeding Series. Proceedings of the 6th International Conference on Web Engineering (ICWE 2006)*. Ed. ACM. pp. 281–288. 2006.
15. OMG: MDA Guide, <http://www.omg.org/docs/omg/03-06-01.pdf>. Version 1.0.1. 2003.
16. N. Moreno, P. Fraternali and A. Vallecillo. A UML 2.0 profile for WebML modelling. *II International Workshop on Model-Driven Web Engineering*. Palo Alto, California. 2006.
17. MDWE Workshop. <http://mdwe2008.pst.ifi.lmu.de/>.
18. OMG. *Unified Modeling Language: Superstructure*, version 2.0. Specification, OMG, 2005. <http://www.omg.org/cgi-bin/doc?formal/05-07-04>.
19. G. Rossi. An object-oriented method for designing hypermedia applications. PhD Thesis. University of PUC-Rio. Rio de Janeiro. Brazil. 1996.
20. Schauerhuber, A., Wimmer, M. and Kapsammer, E. 2006. Bridging existing web modeling languages to model-driven engineering: a metamodel for WebML. *2nd International Workshop on Model-Driven Web Engineering*, Palo Alto, California.
21. SDMetrics. <http://www.sdmetrics.com/>.

Aligning Business Motivations in a Services Computing Design

T. Roach, G. Low, and J. D'Ambra

Abstract The difficulty in aligning business strategies with the design of enterprise systems has been recognised as a major inhibitor of successful IT initiatives. Service-oriented architecture (SOA) initiatives imply an entirely new approach to enterprise process enablement and require significant architectural redesign. Successful SOA deployments are highly dependent on the degree to which flexible support for evolving business strategies is embedded into their designs. This chapter addresses the challenge of modelling business strategies in support of SOA designs. The proposed framework is derived from conceptual elements introduced in the OMG business motivation model and delivers an architectural view for business stakeholders in a computational-independent model (CIM). This model represents the first of three layers that will define a complete reference architecture for a service-based computing model.

Keywords Service oriented architecture · Enterprise systems · Business strategies · Service-based computing model

1 Introduction

Modern enterprise applications promise dramatic improvements in corporate productivity and efficiency; however, the rapid assimilation of these evolving technologies has come at a cost. Most large enterprises possess a myriad of business applications and information resources built on diverse technology standards, precariously tied together through a similarly diverse array of interfaces. As these complex business systems have proliferated, IS teams have faced the ongoing challenge of maintaining a consolidated description of their enterprise systems; how they are accessed, what interdependencies exist and how the information they manipulate is managed [1–3]. Considerable effort has been applied in modelling and describing

T. Roach (✉)

School of Information Systems, Technology and Management, University of New South Wales, Sydney, NSW, Australia

e-mail: t.roach@student.unsw.edu.au

these complex application environments, not only to document how the enterprise systems are organised but also to clarify the flow of processes and information and to establish guidelines and principles for extensions and enhancements to the system [4–6].

These consolidated abstractions of system environments have come to be known as Enterprise Architectures [7]. The Enterprise Architecture description could be considered the highest level abstraction of the entire systems environment of an organisation. It provides a multidimensional blueprint of the arrangement of software applications and information technologies and how they work together [7–13]. One of the principal objectives of an Enterprise Architecture description is to serve as a reference for facilitating communication between the various stakeholder groups over technical designs. This is typically achieved through abstract representations of the system in the form of architectural views which provide a perspective of the system in a context that is meaningful to that stakeholder's interest [14]. The principle issue is to facilitate understanding between technical and business stakeholders each of whom use entirely different language and have very different motivations [15].

Enterprise Architecture frameworks have had some success at delivering comprehensive architectural perspectives for technical stakeholders, but have not been particularly successful in providing well-defined business perspectives of enterprise systems [16–20]. Where business strategy is contemplated in existing frameworks, it is generally used as input into architectural design decisions, but is not integrated into the architectural description itself. For instance, a common criticism of Zachman is that his model has very limited support for integrating business motivations into the technical designs [5, 6].

This difficulty in aligning business and IT perspectives on enterprise systems has been well recognised and researched, with many approaches suggested for addressing the problem (see Chan and Reich for a comprehensive review of alignment research in IS) [21, 22]. Business users of IT systems commonly complain of the enormous challenge in extending and adapting technology solutions to address evolving business needs [2]. The typical result is that as new needs emerge, they are addressed through a proliferation of niche business applications based on diverse technology platforms, giving rise to redundant capabilities, replicated information and complex system interfaces to tie the disparate applications together. The resulting “silo-based” architecture is a well-recognised reality that many large organisations are struggling to maintain [15, 23].

A recent wave of optimism has emerged for addressing this problem through the growing momentum behind service-oriented architectures (SOA), a distributed computing approach to designing flexible business systems around loosely coupled web services. SOA designs hold enormous potential for improved flexibility in how we access and make use of information technology and for removing application silos by breaking down application logic into loosely coupled reusable services that are orchestrated on-the-fly into context-aware business processes [24–28]. The SOA approach has received huge publicity and is widely touted as the new paradigm for enterprise computing. The technical concepts behind services computing are

not difficult to understand. However, adopting an enterprise-wide SOA philosophy is an extremely complex undertaking. This complexity is largely a function of the breadth of business concerns that need to be considered in the design and governance of the architecture. Since the ultimate intention is to support service, process and data definitions that have enterprise-wide relevance, clarity for all stakeholders around the architectural design and definitions becomes fundamentally important [28].

While the technical benefits of SOA designs are reasonably well understood, some scepticism remains about the business benefits [29]. The success or failure of SOA projects will largely depend on the ability of business stakeholders to fully understand its potential and for SOA architects to effectively incorporate business strategies into a closed loop architectural design. Recognition of this challenge has sparked a renewed interest in reference frameworks for coordinating alignment between business requirements and technical designs for SOA designs. A reference architecture for an enterprise system should provide the framework for this articulation of business motivations in such a way that technical designs can be clearly mapped to business requirements and that there is intrinsic linkage between the business strategies and the processes, services and information needed to support those strategies [30]. This chapter presents the first of three layers that will define a complete reference architecture for business and IT alignment that has particular relevance for a service-based computing model. The chapter is organised as follows: relevant business motivation literature is reviewed in Section 2, the proposed model is presented in Section 3 while Section 4 presents the conclusions.

2 Business Motivation Modelling

The activity of business motivation modelling is concerned with establishing the semantics for appropriately describing the Business Purpose of an organisation and the mechanisms for achieving that purpose. This field has evolved from early attempts to establish a structure for the way in which business rules are defined with the intention of developing an accepted taxonomy for the semantic definitions of the core concepts used in strategic business planning [31].

The elaboration and implementation of business strategies are a widely researched field [32–35]. At least two significant streams of thought have been embraced by the practitioner community for documenting and communicating business strategies in practice. Kaplan and Norton have made an enormous contribution in this field with strategy maps, a concept which evolved out of their balanced scorecard approach to measuring business performance [36]. Strategy maps provide a simple one-page structure for aligning strategic objectives in an explicit cause-and-effect relationship with one another across the four traditional balanced scorecard perspectives (financial, customer, internal, learning and growth). Strategy maps have proven to be highly effective tools for communicating and implementing business strategies, although effective translation of strategy maps into business requirements for technical system designs remains largely unexplored.

A second approach to defining a taxonomy and semantics for strategic business planning has been developed by the Object Management Group (OMG) with their business motivation model (BMM) [31]. The Object Management Group is dedicated to developing standardised modelling notations that reduce complexity and improve interoperability in software design. OMG is an independent, not-for-profit organisation that has broad industry support and has contributed several well-known industry standards including CORBA, DDS, UML, MOF, MDA and BPMN amongst other evolving standards [37].

BMM has evolved from early drafts developed by the Business Rules Group (BRG) in 1995. OMG first adopted the BMM standard in 2005 and in 2007 published the current version, release 1.3. This work is currently being extended with further standards pending finalisation for a business process modelling notation (BPMN), business process definition metamodel (BPDm) and the business process maturity model (BPMm).

BMM is a metamodel that identifies standard descriptions for the elements of a Business Purpose. The key components of the BMM are arranged into two main areas namely the organisation ends (goals and objectives) and means (strategies and tactics, governed by policies and rules). It is important to note that BMM is not intended to provide a complete business model. BMM defines the motivations for the business existing, but it does not establish the boundaries for the environment or scope in which the business operates, nor does it address how business initiatives are implemented in support of the Business Purpose; for example, how courses of action are actually executed in the form of business initiatives. There is no assignment of activities to roles and there are no definitions of the attributes that establish the business vocabulary or any of the dynamics or interdependencies between business participants and business resources.

3 Defining a Business View for a Reference Architecture

The objective of this chapter is to propose a framework for extending the business motivation model to provide a more complete business context. The outcome is a comprehensive *Business View*, the first step towards a complete enterprise reference architecture.

OMG recognises that BMM needs to be inserted into a broader framework to provide a complete business model [31] and in fact BMM makes reference to some of these other business concepts that need to be considered in a comprehensive business model. The BMM concept of influencer addresses the roles that participants, organisation units and resources play in a way the business is executed. These elements provide definitions for the human and physical resources that participate in the business plan. They begin to define the *Business Scope*; the environment in which the business will operate. BMM also contains a placeholder in their metamodel for business process and makes reference to the fact that processes, workflow and a business vocabulary are essential elements of a full business model. These elements describe the *Business Implementation*; the mechanism that determines the way in which the business will operate.

The proposed *Business View* will adopt the principal components of BMM as the *Business Purpose*. It will extend the BMM model by providing additional definitions for the *Business Scope* and the *Business Implementation*. The *Business View* represents a computational-independent model (CIM) as defined by OMG’s model-driven architecture (MDA) [38]. Later research will complete the reference architecture, mapping the *Business View* to a *Technical View* representing a platform-independent model (PIM) and finally developing a *Platform View* representing a platform-specific model (PSM) to complete the reference architecture.

The constructs of a model can be likened to the structure of language. A model communicates a concept in much the same way that a sentence communicates a message. In language we recognise that for sentences to be understood they need to conform to certain rules or patterns for their construction. A sentence has syntactical *Structure* in its grammar, it has semantic *Meaning* in the message being communicated and to be fully understood the receiver of the message needs to have an unambiguous understanding of the *Context* in which the sentence is stated. These three requirements are essential characteristics for effectively communicating a message. The design of a model is essentially intended to communicate a complex message; a comprehensive representation of the subject being modelled. To achieve an accurate representation, the message must be well understood and should therefore conform to the requirements for communicating any message. All of the dimensions in the proposed reference architecture will be defined in terms of their Context, Structure and Meaning.

The proposed *Business View* will define the Context, Structure and Meaning of the Business Purpose, Scope and Implementation, Fig. 1.

Defining the Business Purpose: The *Business Purpose* describes its reason for existing. It establishes the business vision and mission as well as the governance model for achieving the vision. The OMG business motivation model provides a well-accepted framework for defining the *Business Purpose*. The proposed *Business View* adopts the BMM constructs for expressing the Business Purpose. These constructs can be mapped to the *Business View* (purpose layer of Fig. 5) as follows:

- *Context:* desired results (goals and objectives) position the context for what the *Business Purpose* is intended to achieve. The Context defines measurable outcomes for the courses of action.

	Meaning	Structure	Context
Business Purpose		Why and When Questions	
Business Scope		Where and Who Questions	
Business Implementation		What and How Questions	

Fig. 1 Business View CIM

- *Structure*: directives (policies and rules) enforce governance over the execution of courses of action. This governance provides a clear structure within which the *Business Purpose* will be executed.
- *Meaning*: courses of action (strategies and tactics) determine the plan for delivering on the goals and objectives. They give meaning to the *Business Purpose* by defining the way in which the desired results will be achieved.

The *Business Purpose* addresses the “*Why*” questions for the Business View; it defines why the business exists through clear elaboration of its vision and mission (see Fig. 1). Because goals and objectives must be measurable, they are by definition time bounded, establishing target timeframes for achieving the desired outcomes. Goals and objectives therefore also address the “*When*” questions of the business plan and are constantly assessed against the attainment of those targets over time.

BMM also provides a detailed metamodel for defining the relationships between the Business Purpose concepts which are summarised in Fig. 2.

Defining the Business Scope: Having determined the purpose of the business, the Business View needs to establish the boundaries within which the business will operate or in other words, the Business Scope. The Business Scope is concerned with the deployment of resources in pursuit of the Business Purpose. It establishes how the influencers will channel effort towards the Business Purpose through the roles and relationships in the organisation structure. It describes the commitments the business will undertake and the allocation of responsibilities for those commitments.

The Business Scope defines the “*Who*” questions relating to the stakeholders in the business. It also addresses the “*Where*” questions relating to the sphere of operation of the business and the locations of assets and operations. The Business Scope (scope layer of Fig. 5 and associated metamodel, Fig. 3) describes

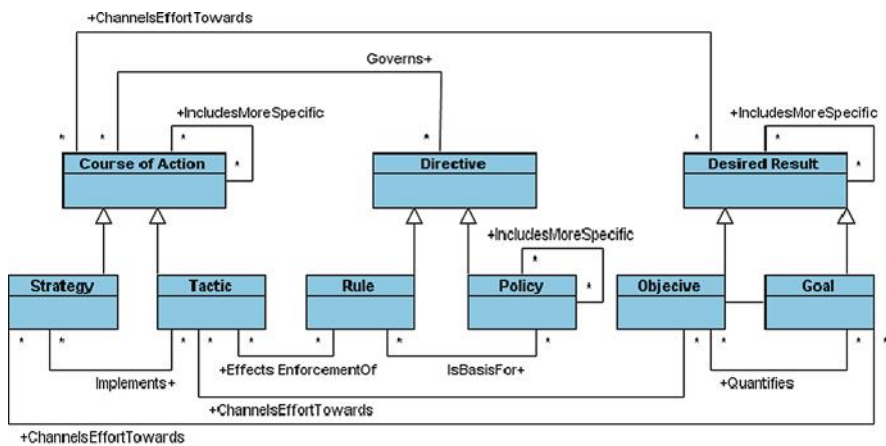


Fig. 2 Business Purpose metamodel

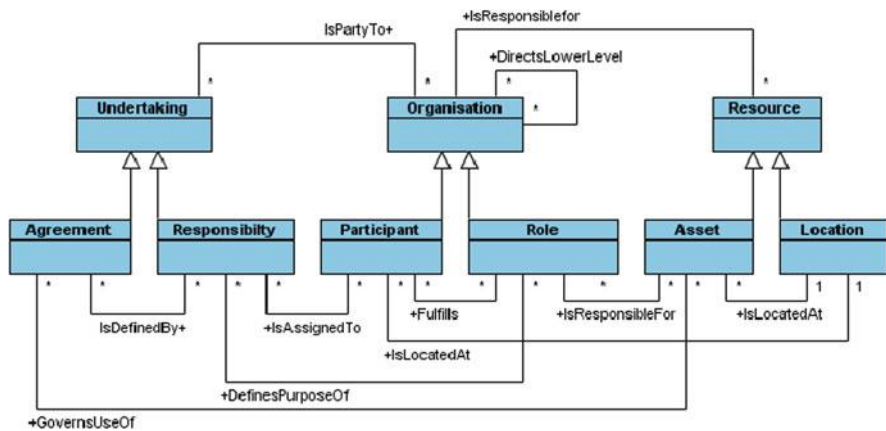


Fig. 3 Business Scope metamodel

- *Context*: resources (assets and locations) are physical assets that are deployed in pursuit of the Business Purpose. Assets include, for example, capital assets such as sites, equipment and raw materials as well as the inventory of products that the business produces. The deployment of resources involves the allocation of assets across the geographical locations of the business sites. Resource definitions provide context for the sphere of operation in the Business Scope.
- *Structure*: organisation (participants and roles) provides the structure for the alignment of stakeholders in pursuit of the Business Purpose. Stakeholders include internal human resources such as individual participants as well as their organisational units. Stakeholders also include external participants such as partners, suppliers and customers. The organisation identifies the participants in the *Business Scope*, defines their roles and maps out the structure of their relationships.
- *Meaning*: undertakings (agreements and responsibilities) involve the establishment of agreements relating to the allocation of responsibilities. Undertakings give meaning to the *Business Scope* by establishing responsibilities for participants in the business as well as for the deployment of company resources. Undertakings define responsibilities for internal participants (e.g. the obligations of individuals and organisation units), responsibilities for assets (e.g. service agreements and product warranties) as well as responsibilities in relation to external participating organisations (e.g. supplier contracts or customer orders).

Defining the Business Implementation: The *Business Implementation* describes the business operations conducted by the business in pursuit of its purpose. Implementation is concerned with putting the plan into effect. It describes the mechanisms for the deployment of resources and participants.

The *Business Implementation* definition articulates the Context, Structure and Meaning for “How” the Business Purpose will be achieved within the defined

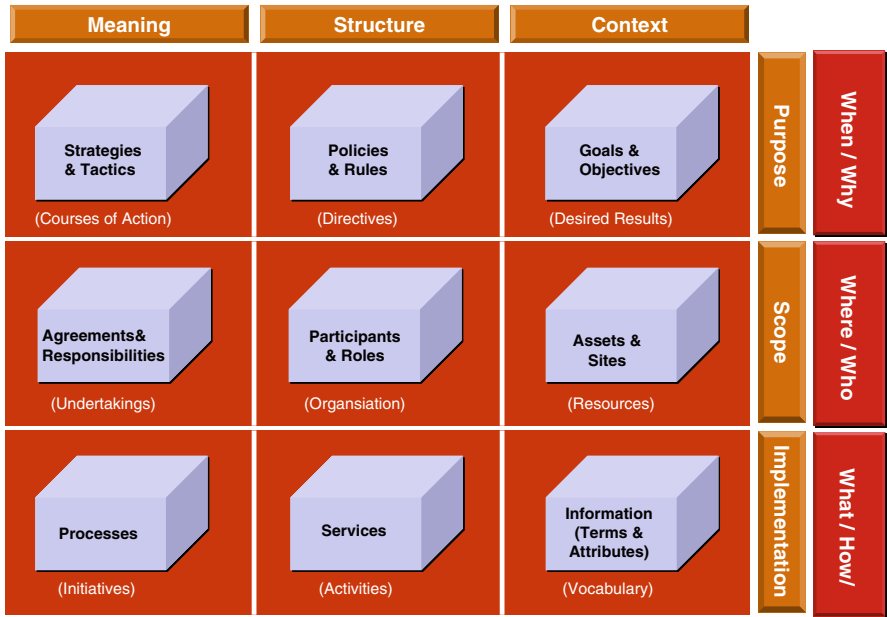


Fig. 5 Consolidated Business View

4 Consolidating the Business View

The consolidated *Business View* is an integrated model of Purpose, Scope and Implementation, aligning the constructs of these three domains and establishing the relationships between them, illustrated in Fig. 5. The consolidated *Business View* is expressed in the *Business View* metamodel shown in Fig. 6.

The research goal is to design a *Reference Architecture for Enterprise Computing*. The current momentum towards loosely coupled, dynamic architectural styles reiterates the need for theoretical models that can assist stakeholders gain a comprehensive architectural perspective that supports service-based principles. This chapter outlines the first layer (Business View) of a reference architecture to support this demand. Validation of the model in a real-world situation is still pending.

The *Business View* describes the first and most important component of the reference architecture. Elaborating a comprehensive model of the business is the essential first step for planning a migration to a service-based computing platform. The proposed *Business View* has been designed with forethought into the requirements of the underlying *Technical View* that it needs to align with. In particular, the business definitions for processes, services and information in the *Business View – implementation* domain will greatly facilitate the technical models of enterprise processes, the governance of enterprise services and the mastering of enterprise information in a SOA implementation.

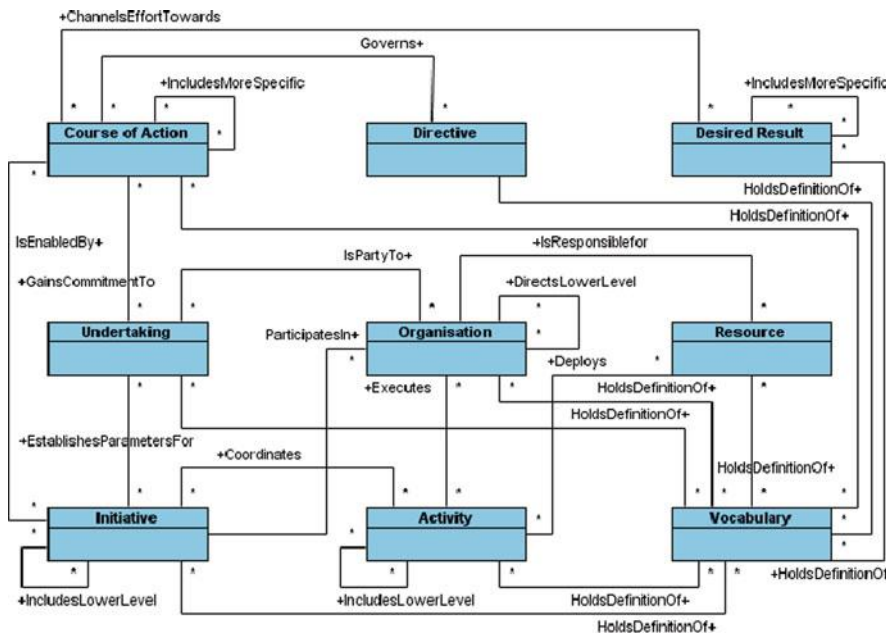


Fig. 6 Business View metamodel

The unique contribution of this model is the approach of designing a reference architecture that is explicitly derived from business strategies. As mentioned in the introduction, mainstream Enterprise Architecture frameworks have suffered criticism for their overtly technical focus and lack of alignment with business concerns [5, 6]. The Zachman framework, for example, has only three cells under “*motivation*” which address the entire Business View as described above, namely *List of Business Goals/Strategies*, *Business Plan* and *Business Role Model* [12]. However, no further guidance is provided for the elaboration of these concepts and neither is any explicit relationships defined between these business concerns and the associated technical constructs in the framework. The Business View defined in this chapter provides a comprehensive model of the business from a business stakeholder perspective that provides a solid foundation for strategic alignment with the explicitly related underlying technical designs and platform deployment.

5 Conclusion

Aligning business strategy with the technical designs of information systems is a challenging endeavour and one that organisations have been struggling with for decades. Substantial effort has been applied in developing models and frameworks for describing Enterprise Architectures, but the focus of these efforts has been very much concentrated on modelling the technical perspective of the applications and technology platforms that support large organisations. Integrating a

technical architecture framework into a comprehensive business model is an area of Enterprise Architecture research that is largely unaddressed and this lack of a unified reference architecture represents a major gap in IS research. OMGs business motivation model represents a starting point for such an exercise, but does not deliver a complete business model. The *Business View* proposed in this chapter provides a complete perspective of the business concerns that need to be defined as the first step in designing a *Reference Architecture for Enterprise Computing*. The proposed *Business View Metamodel* provides the detail for a computational-independent model as a first step in the design of such a reference architecture.

The evolving *Reference Architecture for Enterprise Computing* is based on the OMG theory of model-driven architecture. The Business View elaborated above represents a computational-independent model, as defined in MDA. The framework for the *Business View* (Context Scope and Meaning; Purpose, Structure and Implementation) will be applied to the development of two more models, namely a *Technical View* and a *Platform View*. Whereas the *Business View* has focused on business definitions, the *Technical View* will focus on technical designs and the *Platform View* will focus on platform deployment.

References

1. Dehning, B., Richardson, V.J. and Zmud, R.W. (2003) The value relevance of announcements of transformational informational technology investments, *MIS Quarterly*, 27(4), 637–656.
2. Feld, C.S. and Stoddard, D.B. (2004) Getting IT right, *Harvard Business Review*, 82(2), 72–79.
3. Luftman, J. (2003) Assessing IT/business alignment, *Information Systems Management*, 20(4), 9–15.
4. Strnadl, C.F. (2006) Aligning business and IT: the process-driven architecture model, *Information Systems Management*, 23(4), 67–77.
5. Braun, C. and Winter, R. (2005) A comprehensive enterprise architecture metamodel and its implementation using a metamodeling platform. In Desel, J. and Frank, U. (eds), *GI-Edition Lecture Notes in Informatics (LNI)*, pp. 64–79. Klagenfurt, Austria: Gesellschaft für Informatik.
6. Jonkers, H., van Burren, R., Arbab, F. et al. (2003) Towards a language for coherent enterprise architecture descriptions. In *Proceedings of the Seventh IEEE International Enterprise Distributed Object Computing Conference (EDOC'03)*, Brisbane, Queensland, Australia, September 16–19, pp. 28–37.
7. Zachman, J.A. (1987) A framework for information systems architecture, *IBM Systems Journal*, 26(3), 276–292.
8. AGIMO (2007) *Australian Government Architecture*. Retrieved May 18, 2009 from: <http://www.finance.gov.au/e-government/strategy-and-governance/australian-government-architecture.html>.
9. CapGemini (1993) *Integrated Architecture Framework*. Retrieved May 18, 2009 from: http://www.capgemini.com/services/soa/ent_architecture/iaf/.
10. DoDAF (2007) *US Department of Defense Architecture Framework (DoDAF)*, V1.5, Vols I, II, and III.
11. FEA (2009) *Federal Enterprise Architecture Program Management Office*. Retrieved May 18, 2009 from: <http://www.whitehouse.gov/omb/e-gov/fea/>.
12. Sowa, J.F. and Zachman, J.A. (1992) Extending and formalizing the framework for information systems architecture, *IBM Systems Journal*, 31(3), 590–616.

13. TOGAF (2006) *The Open Group Architecture Framework Version 8.1.1*. Retrieved May 18, 2009 from: <http://www.opengroup.org/togaff/index811.htm>.
14. IEEE (2000) IEEE recommended practice for architectural descriptions of software intensive systems, *IEEE Standards* 1471–2000.
15. Ross, J.W., Weill, P. and Robertson, D.C. (2006) *Enterprise Architecture as Strategy*. Boston, MA: Harvard Business School Press.
16. Lindström, Å., Johnson, P., Johansson, E., Ekstedt, M. and Simonsson, M. (2006) A Survey on CIO concerns-do enterprise architecture frameworks support them? *Information Systems Frontiers*, 8(2), 81–90.
17. Infosys (2005) *Enterprise Architecture Survey 2005*, Infosys Technologies Limited.
18. Infosys (2007) *Enterprise Architecture Survey 2007 – Enterprise Architecture is Maturing*, Infosys Technologies Limited.
19. Schekkerman, J. (2005) *Trends in Enterprise Architecture 2005: How are Organizations Progressing?* Institute for Enterprise Architecture Developments (IFEAD).
20. Sessions, R. (2007) *A Comparison of the Top Four Enterprise Architecture Methodologies*. Retrieved May 18, 2009 from: <http://msdn.microsoft.com/en-us/library/bb466232.aspx>.
21. Chan, Y.E. and Reich, B.H. (2007) IT alignment: an annotated bibliography, *Journal of Information Technology*, 22(4), 316–396.
22. Chan, Y.E. and Reich, B.H. (2007) IT alignment: what have we learned?, *Journal of Information Technology*, 22(4), 297–315.
23. Weill, P. and Aral, S. (2006) Generating premium returns on your IT investments, *MIT Sloan Management Review*, 47(2): 39–48.
24. Abrams, C.A. and Andrews, W. (2005) *Client Issues for Service Oriented Business Applications*, Gartner Research Number G00127943.
25. Austvold, E. (2005) *Service-Oriented Architectures: The Promise and the Challenge*, AMR Research.
26. Erl, T. (2005) *Service-Oriented Architecture (SOA): Concepts, Technology, and Design*, Upper Saddle River, NJ: The Prentice Hall Professional Technical Reference.
27. Hart, T (2004) *Enabling the Service Oriented Enterprise*, DataMonitor.
28. Merrifield, R., Calhoun, J. and Stevens, D. (2008) The next revolution in productivity, *Harvard Business Review*, 86(6), 72–80.
29. Havenstein, H. (2006, August 7) Proving SOA worth is a big challenge for IT: tools emerging to manage, measure benefits of the complex architecture, *Computerworld*, 40, 1–7.
30. Roach, T., Low, G. and D'Ambra, J. (2008) CAPSICUM – a conceptual model for service oriented architecture. In *Proceedings – 2008 IEEE Congress on Services (SERVICE 2008)*, Honolulu, Hawaii, July 6–11, pp. 415–422.
31. OMG (2007), *Business Motivation Model Version 1.0 (Align)*, Object Management Group Document Number: formal/2008-08-02.
32. Mintzberg, H. (1994) *The Rise and Fall of Strategic Planning: Reconceiving Roles for Planning, Plans, Planners*, New York: Free Press.
33. Porter, M.E. (1996) What is Strategy?, *Harvard Business Review*, 74(6), 61–78.
34. Steiner, G.A (1979) *Strategic Planning: What Every Manager Must Know*, New York: Free Press.
35. Tregoe, B.B. and Zimmerman, J.W. (1980) *Top Management Strategy: What it is and How to Make it Work*, New York: Simon and Schuster.
36. Kaplan, R.S. and Norton, D.P. (2004) *Strategy Maps: Converting Intangible Assets into Tangible Outcomes*, Boston: Harvard Business School Press.
37. OMG (2008) *Object Management Group Website*. Retrieved May 18 from: <http://www.omg.org/>.
38. OMG (2003) *Model Driven Architecture Guide Version 1.0.1*, Object Management Group Document Number: omg/2003-06-01.

Part V
Information Systems for Service
Marketing and e-Businesses

CRank: A Credit Assessment Model in C2C e-Commerce

Zhiqiang Zhang, Xiaoqin Xie, Haiwei Pan, and Qilong Han

Abstract An increasing number of consumers not only purchase but also resell merchandise through C2C web sites. One of the greatest concerns for the netizens is the lacking of a fair credit assessment system. Trust and trustworthiness are crucial to the survival of online markets. Reputation systems that rely on feedback from traders help to sustain the trust. And reputation systems provide one of the ways of building trusts online. In this chapter, we investigate a credit assessment model, CRank, for the members in the context of e-market systems, such as Alibaba, eBay, to solve such problem as how to choose a credible business partner when the customer wants to purchase some products from the Internet. CRank makes use of feedback profile made up of ranks from other users as well as an overall feedback rating for the user based on the idea of PageRank. This model can be used to build a trustable relation network among business participants.

Keywords Assessment model · e-commerce · Reputation system · Credit

1 Introduction

Online commerce markets and online auction markets are very thriving nowadays. According to an ACNielsen study on Global Consumer Attitudes Towards Online Shopping in February 2008, there are more than 875 million people who claimed that they have shopped online. The number of the online consumer auction sales will reach \$65 billion by 2010, as revealed in the latest report about online auctions by Forrester Research, such number accounts for nearly one-fifth of all online retail sales. In eBay, the largest online auction site in the world, of its 180.6 million registered users, there are 71.8 million active registered users, which takes up 40% of its total number. A critical reason for the success of online auction sites is the use

Z. Zhang (✉)

College of Computer Science & Technology, Harbin Engineering University, Harbin, China
e-mail: zqzhang@hrbeu.edu.cn

of an online feedback system as a reputation system to help sustain trust in online markets [1–8].

Despite of the thriving of online auction and online commerce markets, the current online reputation systems are not perfect. And there are an increasing number of the online auction frauds. As it is known, both the online commerce markets and online auction markets have the essential features of remote trade: the buyers and sellers know nothing about each other and are remote and anonymous to one another until they see each other online. Such features are especially true for the online auction markets. In the remote trade, it is very easy to exit and enter these markets by changing online identities due to the fact that transactions tend to be geographically diffused, which also result in many auction frauds.

Due to the imperfectness of the current online reputation systems, online auction fraud still accounts for a significant proportion of the complaints filed with the FTC. The online reputation systems still have the following major problems including low incentives for providing feedback, bias toward positive feedback, abuse of the reputation system, traders changing identities, and poor credit assessment model. In addition, if the sellers' information about past transaction history is not accurately recorded, good sellers might be driven out of the market akin to the lemon car market of Akerlof [9]. In result, the trust and trustworthiness in online markets would be destroyed by the above problems and thus limit the growth of the online markets.

To solve the problems arose from the credit assessment model, this chapter aims not only at investigating a credit assessment model, CRank, for members in such context of e-market systems as Alibaba, eBay, etc., but also at solving the problems of how to choose a credible business partner when the customer wants to buy some products from the Internet. Meanwhile, basing on PageRank [10] and the business history of e-market, this chapter also proposes an algorithm for evaluating the credit of members.

The rest of the chapter consists of the following sections. In Section 2, we review the literature on reputation mechanism models. In Section 3, we explain the details of CRank model. In Section 4, we introduce a numerical example of the CRank model. In Section 5, the conclusion and possible applications are provided.

2 Related Work

In general [8], the adverse selection model and the moral hazard model are the two types of asymmetric information models. In the setting of adverse selection models, taking bilateral trade as an example, the game begins by choosing the sellers' type (e.g., some sellers are more capable or honest than others), unobserved by buyers. Then an agreed contract was reached both by a seller and a buyer. And the seller acts on his type. For example, products are consistently packed and shipped carefully by careful sellers, whereas careless sellers often fail to deliver the product or the products are consistently advertised according to the true conditions by honest sellers whereas dishonest sellers often fail to provide the true condition of the product. In the setting of moral hazard model, a buyer and a seller begin with symmetric

information and agree upon a contract, but then the seller takes an action unobserved by the buyers (e.g., the seller has an incentive to undercut the quality of a product to maximize his profit).

Different roles are exerted by reputation mechanisms in each of these two settings. In an adverse selection setting, as pointed out by Dellarocas [11], the reputation mechanism helps the community learn such attributes of community members as their ability, honesty, etc. which are initially unknown to the community. While in a moral hazard setting, the threat of future punishment (e.g., in the form of lower bids following the posting of a negative rating on a trader's reputation profile) is reminded by the reputation mechanisms which can promote cooperative and honest behaviors among self-interested economic agents. As Cabral [12] summarizes, typical reputation mechanism models that create "reputation" (i.e., when agents believe a particular agent to be something) are based on Bayesian updating of beliefs and, in an adverse selection setting, possibly signaling [13, 14]. In a moral hazard setting, other essential reputation models are also provided in which "trust" (i.e., when agents expect a particular agent to do something) is created through repeated interaction and the possibility of "punishing" off-equilibrium actions [15–17]. The other related surveys on trust and reputation are also provided by MacLeod [18] and Jøsang et al. [6].

In an online market, it is the reputation system that influences the practice of the traders to behave cooperatively. Besides, the content and format of aggregated information the online reputation system publishes are also determined by the online reputation system itself. Thus, the design of an incentive mechanism that will elicit truthful feedback is really critical to us. And there much work has been done on the literature related to the topic of mechanism design. Some overviews on reputation mechanisms have been carried out by Resnick et al. [19] and Dellarocas [20]. And the other literature work is mainly focused on the efficiency of online reputation mechanism [21]. In order to solve the untruthful reporting problems, others design mechanisms to deal with bad mouthing and ballot stuffing [7, 22].

The designs of mechanisms to induce sellers to behave cooperatively are also suggested in several papers. But some researcher goes further, Ba et al. [23] suggests the trusted third party (TTP) mechanism in which the certificates are issued to both the sellers and the buyers. Dellarocas [24] also proposes that if a seller announces expected quality, he would be charged a listing fee contingent and rewarded for his announced quality and the rating posted for that seller by the winning bidder for that listing. The peer prediction method proposed by Miller et al. [25] is that the feedback providers are awarded by the center (online market). The work done by Dellarocas and Wood [26] is that in order to repair distortions introduced by reporting bias, they provide a sophisticated computational mechanism. Though these mechanisms are impressive, yet each has some disadvantages. If we compare the two mechanism provided by Ba et al. and Dellarocas, we may find that Ba et al.'s [5] mechanism can induce cooperative behavior if both buyers and sellers obtain the verifications from the TTP, but extra transaction cost is imposed to buyers. In contrast, Dellarocas's [24] mechanism requires all buyers to report. As far as the work of Miller et al.'s [25] is concerned, he requires the market maker to provide incentives to rating providers.

Finally, both Dellarocas and Wood's [26] requires buyers to consider taking the missing feedback, though such information is very difficult to access for an average buyer.

In the work of Lingfang Ivy Li's [8], a mechanism to induce buyers to report was proposed. It shows that giving sellers an option to provide a rebate to cover the reporting cost to a buyer. This will provide an incentive for buyers to provide feedback without any loss to the online market maker. However, the cost an online auction market might have is to produce this option for sellers, which may not be substantial.

In this chapter, we focus on how to utilize the business log in e-market or collect data from the third party, such as the online payment system. We propose a CRank algorithm for evaluating the credit of sellers. Comparing with the above methods, CRank method pays more attention to the trust links among all business participants. Furthermore, this method can cooperate with the above mechanisms in building up more perfect reputation model. In addition, our model is more simple and easy to implement.

3 Credit Evaluation Model

In a typical C2C community, such as www.taobao.com, the customer must be the community member before he/she starts a business with other members in this community.

If one buyer wants to purchase goods, he/she could use classical keyword-based methods for searching commercial goods. Then, the system will return a seller list where the system ranked the candidate business partners based on some principles. One of the main problems that plague modern C2C systems is the lack of flexible credit measurement for all members. In many situations, C2C platforms retrieve hundreds of sellers that all are able to provide the goods which at least partially satisfied the input query.

Trust and trustworthiness are crucial to the survival of online markets, and reputation systems that rely on feedback from traders help sustain this trust. In this section, we will introduce a credit evaluation model.

3.1 Basic Concepts

Suppose there are n members in the C2C system. They do business each other. Both business participants will give an evaluation to another business partner with a numerical score. E-market or third reputation system will log all the business information in database.

We can use an undirected graph to model the business state of e-market in period $[T_0, T]$ as following:

$$G = (V, E, \omega)$$

Business graph G is a 3-tuple, where V is a finite set of nodes, and $V = \{M_i | M_i \text{ is the member of system}\}$; E is a finite set of edges and $E = \{(M_i, M_j)^{\text{BID}} | \text{if } M_i \text{ did a business BID with } M_j \text{ in period } [T_0, T], M_i \text{ is the buyer and } M_j \text{ is the seller, where BID is the identifier of this business}\}$; ω is a weight function from E to a value pair set. For example, if M_1 did a business with M_2 , and M_1 's evaluation score for M_2 is 4, and M_2 's evaluation score for M_1 is 3, then we have $\omega(M_1, M_2)^{\text{BID}} = (4, 3)$.

Obviously, there is more than one edge between two nodes, because there is more than one business occurred between two members.

Simplifying the discussion, we define another graph G' from the graph G :

$$G' = (V', E', \omega')$$

where $V = V'$, $E' = \{(M_i, M_j) | \text{if there is at least one business transaction between } M_i \text{ and } M_j, i < j \text{ and } i \neq j\}$, ω' is a weight function from E' to a value pair set. For example, if M_1 did two businesses with M_2 , and M_1 's evaluations for M_2 is 4 and 3, respectively, and M_2 's evaluations for M_1 is 5 and 4, respectively, then

$$\omega'(M_1, M_2) = (4 + 3, 5 + 4) = (7, 9)$$

Comparing with graph G , G' is a more simplified version which compress multi-edges between two nodes in G into one edge in G' . For the above example, there are two edges between M_1 and M_2 in graph G , and there is only one edge in graph G' .

Matrix $A = \{a_{ij}\}$ is a business matrix, where a_{ij} is the total number of business in period $[T_0, T]$ between M_i and M_j and $a_{ii} = 0$, $i = 1, \dots, n$. Matrix A is a symmetric matrix. T_0 is any specified starting time and $T_0 < T$.

Matrix $B = \{b_{ij}\}$ is an evaluation matrix, where

$$b_{ij} = \begin{cases} 0, & i = j \\ \omega'(M_i, M_j)_{[1]}, & i \neq j \end{cases}$$

and $\omega'(M_i, M_j)_{[1]}$ is the first element of pair $\omega'(M_i, M_j)$. In other words, b_{ij} is the sum of evaluating scores from M_i to M_j .

Vector $EV = \{s_1, s_2, \dots, s_p\}$ is an evaluating level vector, where s_i is the numerical number of space R and $s_1 < s_2 < \dots < s_p$, such as $EV = \{1, 2, 3, 4, 5\}$.

3.2 CRank Model

In this section, we will introduce a credit evaluation model, CRank model, based on the idea of PageRank [10] that is "one member's credit was obtained from other

members who did businesses with it.” Similar to the PageRank, we can establish the model based on the graph G' as following:

$$CR(M_i) = (1 - d) + d \sum_{j=1}^n CR(M_j) d_{M_{ji}} \quad (3.1)$$

$$\text{where } d_{M_{ji}} = \frac{b_{ji}}{\sum_{k=1}^n b_{jk}} \quad (3.2)$$

- $CR(M_i)$ – means the credit rank value of member M_i . We call the value of $CR(M_i)$ as CRank of M_i . At the beginning when no business has been done, each member has a notation of its own self-credit. That is “ $CR(M_1)$ ” for the first member in the system all the way up to “ $CR(M_n)$ ” for the last member.
- $d_{M_{ji}}$ – each member spreads its vote out, scale-down among all of its links. The factor for member M_1 is “ $d_{M_{1i}}$,” “ $d_{M_{ni}}$ ” for member M_n , and so on for all members.
- $CR(M_j) d_{M_{ji}}$ – if member M_i has edges from member M_n , the share of the vote that member M_i will get is “ $CR(M_n) d_{M_{ni}}$.”
- $d(\dots)$ – all these fractions of votes are added together, but to prevent the other members having too much influence, this total vote is “damped down” by multiplying it by 0.85 (the factor “ d ”).
- $(1 - d)$ – the $(1 - d)$ bit at the beginning is a bit of probability in math magic. So the “sum of all members’ credit values will be one”: it adds in the bit, lost by the $d(\dots)$. It also means that if a member has no links to it even then it will still get a small credit rank value of 0.15 (i.e., $1 - 0.85$).

First of all, we see that the credit rank value of member M_i is recursively defined by the credit rank value of those members which link to M_i .

The credit rank value of node M_j which links to node M_i does not influence the value of member M_i uniformly. Within the CRank algorithm, the credit rank value of the member M_j is always weighted by the factor $d_{M_{ji}}$ on M_j . This means that the smaller value a member M_j has, the less will member M_i benefit from a link to it on member M_j .

The weighted value of members M_j is then added up. This will result in the conclusion that an additional inbound link for member M_i will always increase member M_i ’s credit rank value.

Finally, the sum of the weighted values of all members M_j is multiplied with a damping factor d which can be set between 0 and 1. Thereby, the extension of CRank benefit for a member by another member’s linking to it is reduced.

3.3 Refinement of the Model

From (3.2), we see that M_j devote its whole credit rank value to other $n-1$ partners. Because of

$$\sum_{i=1}^n d_{M_{ji}} = \sum_{i=1}^n \frac{b_{ji}}{\sum_{k=1}^n b_{jk}} = 1$$

In general, one member does not give full mark to his partner. For example, suppose “5” is the highest rating scores, one member maybe given a “4” to his partner. Even this is good rating for the partner, but it also reveals that the partner’s performance is not perfect in business.

Let us have a look at one example which illustrated a weakness of the model. Suppose there are two members in the system. They finish a business during period of $[T_0, T]$. The business graph is shown in Fig. 1:

$$\omega'(X, Y) = (2, 3), d_{XY} = \frac{2}{2} = 1, d_{YX} = \frac{3}{3} = 1$$

Assume that the initial value is $CR(X) = CR(Y) = 1$.

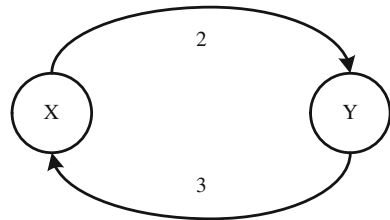


Fig. 1 An example

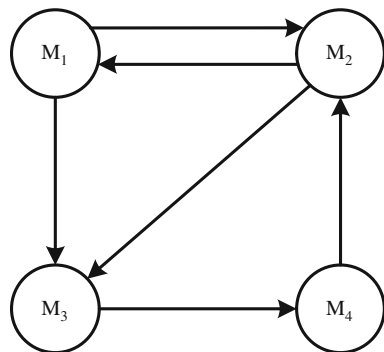


Fig. 2 Graph G of the example

After finishing this business, we can get the following results:

$$CR(X) = 0.15 + 0.85(CR(Y) \times d_{YX}) = 0.15 + 0.85 = 1$$

$$CR(Y) = 0.15 + 0.85(CR(X) \times d_{XY}) = 0.15 + 0.85 = 1$$

The results for X and Y are the same after business. This is unfair because the performance of X and Y are not the same. In fact, X is better than Y . This is not in accordance with the principle “Lower rating and lower credit rank value.” So we refine the model as follows:

$$CR(M_i) = (1 - d) + d \sum_{j=1}^n CR(M_j) b_{ji} c_j \quad (3.3)$$

$$\text{where } c_j = \begin{cases} \frac{1}{s_p * \sum_{k=1}^n a_{jk}}, & \text{if } \exists a_{jk} \neq 0 \\ 0, & \text{otherwise} \end{cases} \quad (3.4)$$

In (3.3), we adjust the spread factor of member. Every member does not devote its whole credit rank value to other members, but devote a discount credit rank value.

For the above example, $b_{XY} = 2$, $b_{YX} = 3$, and

$$c_X = \frac{1}{5 \times 1} = \frac{1}{5}, c_Y = \frac{1}{5 \times 1} = \frac{1}{5}$$

After first iteration, we get the following results:

$$CR(X) = 0.15 + 0.85(CR(Y) b_{YX} c_Y) = 0.15 + 0.85 \times 0.6 = 0.66$$

$$CR(Y) = 0.15 + 0.85(CR(X) b_{XY} c_X) = 0.15 + 0.85 \times 0.66 \times 0.4 = 0.3744$$

4 A Numerical Example

In this section we take an example for illustration purpose. Let us assume that a C2C system has four members and finished six business transactions in period $[T_0, T]$. The log is given below (Table 1). Figure 2 and 3 are the graph G and G' of this numerical example respectively.

From Fig. 3, we get the following results:

$$\omega'(M_1, M_2) = (7, 9), \omega'(M_1, M_3) = (4, 4), \omega'(M_2, M_3) = (5, 5)$$

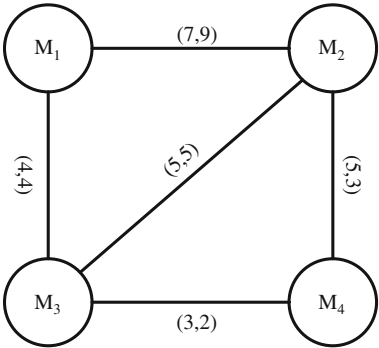
$$\omega'(M_3, M_4) = (3, 2), \omega'(M_2, M_4) = (5, 3)$$

Matrix A and Matrix B of this example are listed below:

Table 1 The log of the C2C system

BID	Buyer	Seller	Rating vector
1	M ₁	M ₂	M ₁ →M ₂ :3, M ₂ →M ₁ :4
2	M ₁	M ₃	M ₁ →M ₃ :4, M ₃ →M ₁ :4
3	M ₂	M ₃	M ₂ →M ₃ :5, M ₃ →M ₂ :5
4	M ₂	M ₁	M ₂ →M ₁ :5, M ₁ →M ₂ :4
5	M ₃	M ₄	M ₃ →M ₄ :3, M ₄ →M ₃ :2
6	M ₄	M ₂	M ₄ →M ₂ :3, M ₂ →M ₄ :5

Fig. 3 Graph *G* of the example



$$A = \begin{pmatrix} 0 & 2 & 1 & 0 \\ 2 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}, B = \begin{pmatrix} 0 & 7 & 4 & 0 \\ 9 & 0 & 5 & 5 \\ 4 & 5 & 0 & 3 \\ 0 & 3 & 2 & 0 \end{pmatrix}$$

After eight iterations, we get the following results (shown in Table 2).
Final results are shown as following:

$$CR(M_1) = 0.452, CR(M_2) = 0.535, CR(M_3) = 0.423, CR(M_4) = 0.336$$

Table 2 Credit value of four members

Iteration	CR(<i>M_i</i>)			
	CR(<i>M₁</i>)	CR(<i>M₂</i>)	CR(<i>M₃</i>)	CR(<i>M₄</i>)
1st	0.759167	0.989469	0.702340	0.479660
2nd	0.687669	0.744085	0.545532	0.400859
3rd	0.558266	0.628266	0.478186	0.364791
4th	0.498687	0.576320	0.447518	0.348546
5th	0.471880	0.552855	0.433694	0.341210
6th	0.459771	0.542264	0.427452	0.337898
7th	0.454305	0.537483	0.424634	0.336403
8th	0.451838	0.535324	0.423361	0.335728

So we could get a ranking list $\{M_2, M_1, M_3, M_4\}$ based on above credit values in descending order.

5 Conclusion

Reputation systems play important roles in today's electronic exchange relationships. They represent one of the principal enabling technologies for online auctions, allowing buyers to assess the trustworthiness of unknown sellers. In this chapter, we proposed a credit assessment model-CRank for the business members based on the idea of PageRank. CRank model makes use of feedback profile made up of comments or ranks from other users as well as an overall feedback rating for the user. This model can be used to build a trustable relation network among business participants. Based on CRank model, we can construct the reputation system which can be used to help buyers avoid becoming fraud victims and help the e-commerce platform, such as Alibaba, eBay, to attract more customers and expand the market.

Acknowledgment This paper was supported by the National Natural Science Foundation of China (Grant No. 60803037 and 60803036), and supported by the National High-tech R&D Program of China under Grant No. 2009AA01Z143, and supported by the Fundamental Research Funds for the Central Universities No. HEUCFZ1010 and No. HEUCF100602.

References

1. R. Paul and R. Zeckhauser. Trust among strangers in internet transactions: empirical analysis of eBay's reputation system. The economics of the internet and e-commerce. In M.R. Baye, editor, *Volume 11 of Advances in Applied Microeconomics*. Elsevier, Amsterdam. pp. 127–157, 2002.
2. G.L.U. Shankar, Venkatesh and F. Sulta. Online trust: a stakeholder perspective, concepts, implications, and future directions. *The Journal of strategic information systems*, 11(3–4): 325–344, December 2002.
3. Y.D. Wang and H. Emurian. An overview of online trust: Concepts, elements, and implications. *Computers in Human Behavior*, 21(3): 105–125, 2005.
4. C. Dellarocas. Reputation mechanism design in online trading environments with pure moral hazard. *Information Systems Research*, 16(2): 209–230, June 2005.
5. S. Ba and P. Pavlou. Evidence of the effect of trust building technology in electronic markets: Price premiums and buyer behavior. *MIS Quarterly*, 26(3): 243–268, September 2002.
6. G.A. Akerlof. The market for lemons: Quality uncertainty and market mechanism. *Quarterly Journal of Economics*, 84(3): 488–500, August 1970.
7. C. Dellarocas. The digitization of word-of-mouth: Promise and challenges of online feedback mechanisms. *Management Science*, 49: 1407–1424, October 2003.
8. L. Cabral. *The Economics of Reputation and Trust. A Primer. Preliminary Draft*, Available at http://pages.stern.nyu.edu/~lcabral/reputation/Reputation_June05.pdf, September 2005.
9. B. Klein and K.B. Leffler. The role of market forces in assuring contractual performance. *Journal of Political Economy*, 89: 615–641, 1981.
10. C. Shapiro. Premium for high quality products as returns to reputations. *Quarterly Journal of Economics*, 98(4): 659–680, 1983.
11. D.M. Kreps and R. Wilson. Reputation and imperfect information. *Journal of Economic Theory*, 27(2): 253–279, 1982.

12. P. Milgrom and J. Roberts. Price and advertising signals of product quality. *Journal of Political Economy*, 94(4): 796–821, 1986.
13. D. Diamond. Reputation acquisition in debt markets. *Journal of Political Economy*, 97: 828–862, 1989.
14. W. Bentley MacLeod. Reputations, relationships and the enforcement of incomplete contracts. CESifo Working Paper Series No. 1730; IZA Discussion Paper No. 1978. Available at SSRN: <http://ssrn.com/abstract=885347>, February 2006.
15. A. Jøsang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2): 618–644, March 2007.
16. P. Resnick, R. Zeckhauser, E. Friedman, and K. Kuwabara. Reputation systems. *Communications of the ACM*, 43(12): 45–48, 2000.
17. C. Dellarocas. Reputation mechanisms. In T. Hendershott, editor, *Handbook on Information Systems and Economics*. Elsevier, Amsterdam, 2006.
18. C. Dellarocas. Analyzing the economic efficiency of ebay-like online reputation reporting mechanisms. In *Proceedings of the 3rd ACM Conference on Electronic Commerce*, pp. 171–179, October 2001.
19. R. Bhattacharjee and A. Goel. Avoiding ballot stuffing in ebay-like reputation systems. *Proceeding of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems*, pp.133–137, 2005.
20. C. Dellarocas. Building trust on-line: The design of robust reputation mechanisms for online trading communities. In M.N. Doukidis, G. and N. Pouloudi, editors, *Information Society or Information Economy? A combined perspective on the digital era*, Chapter: number 95–113. Idea Book Publishing, 2004.
21. S. Ba, A.B. Whinston, and H. Zhang. Building trust in online auction markets through an economic incentive mechanism. *Decision Support systems*, 35(3): 273–286, June 2002.
22. C. Dellarocas. Efficiency through feedback-contingent fees and rewards in auction marketplaces with adverse selection and moral hazard. *3rd ACM Conference on Electronic Commerce (EC-03)*, San Diego, CA, USA, pp. 11–18, June 2003.
23. N. Miller, P. Resnick, and R. Zeckhauser. Eliciting honest feedback: The peer prediction method. *Management Science*, 51(9): 1359–1373, September 2005.
24. C. Dellarocas and C.A. Wood. The sound of silence in online feedback: estimating trading risks in the presence of reporting bias. *Management Science*, 54(3): 460–476, March 2008.
25. L.I. Li. Reputation, Trust, and Rebates: How Online Auction Markets Can Improve Their Feedback Mechanisms, *Journal of Economics and Management Strategy*, 19(2): 303–331, May 2010. Available at SSRN: <http://ssrn.com/abstract=1120881>.
26. S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7): 107–117, April 1998.

Towards Agent-Oriented Approach to a Call Management System

Amir Nabil Ashamalla, Ghassan Beydoun, and Graham Low

Abstract There is more chance of a completed sale if the end customers and relationship managers are suitably matched. This in turn can reduce the number of calls made by a call centre reducing operational costs such as working time and phone bills. This chapter is part of ongoing research aimed at helping a CMC to make better use of its personnel and equipment while maximizing the value of the service it offers to its client companies and end customers. This is accomplished by ensuring the optimal use of resources with appropriate real-time scheduling and load balancing and matching the end customers to appropriate relationship managers. In a globalized market, this may mean taking into account the cultural environment of the customer, as well as the appropriate profile and/or skill of the relationship manager to communicate effectively with the end customer. The chapter evaluates the suitability of a MAS to a call management system and illustrates the requirement analysis phase using i^* models.

Keywords i^* Requirement models · Multi-agent system (MAS) · Call Management Centre (CMC) · Relationship Manager (RM)

1 Introduction

Business telephony needs are either outbound calls to customers (e.g. telemarketing products) or inbound calls (e.g. for customer support, handling sales or enquiries). Companies favour outsourcing their call management to dedicated call management centres (CMC) since they tend to have the latest telephone technology and equipment together with additional value-adding software. The CMC's specialized personnel and training save the client company's time and money. A typical CMC may have a number of corporate clients (e.g. banks, insurance companies) and a few

G. Beydoun (✉)

School of Information Systems and Technology, University of Wollongong, Wollongong, NSW, Australia

e-mail: beydoun@uow.edu.au

thousand *relationship managers* (RM) attending to phone calls to end customers of its corporate clients. The operating cost of a CMC includes the RM salaries and the call costs. The shorter the inbound/outbound calls and the less outbound calls a RM makes to achieve a sale the more profitable a CMC.

CMCs can be hosted anywhere in the world with calls often transferred and routed across countries. The call centre industry is a fast growing industry with the demand for call centre personnel expected to continually grow. Salary and training represent 60–80% of an overall call centre's operations budget [13, 15]. It is imperative that their effort is targeted and effective. In other words, the employee with the most knowledge about a given product, with the most suited communication skills and with the most appropriate availability, is the one who should make or receive a service call to/from an end customer. Matching a RM to a customer can be complicated by the dispersed geographic location of the call centre. A RM and a customer are often in different countries and across different time zones. A RM often requires additional communication skills tempered by cultural and geographic sensitivities in addition to product knowledge. We propose using an intelligent distributed system (known as multi-agent systems) to assist in customer relationship management by routing calls and allocating calling duties to the most appropriate RM (in terms of knowledge/skills and availability) to maximize effectiveness.

The use of software agents in a call management system is highly promising. A multi-agent system (MAS) is a collection of software agents. MASs have been shown to be highly appropriate for the engineering of open, distributed, or heterogeneous systems [5, 19]. We envisage a call management MAS consisting of distributed intelligent agents supporting the RMs and knowledge-based agents monitoring the call centre operation ensuring balanced workload allocation to the agent. These agents would ensure the best match between a customer and a RM and monitor the whole system in terms of customer satisfaction and call throughput per RM.

This chapter is part of a research employing MAS technology aimed at helping a call management centre (CMC) making better use of its personnel and equipment while providing a high-value service to its clients and end customers. The chapter is organized as follows: Section 2 describes related work and gives further details on the call management domain. Section 3 sketches details of a MAS solution for call management and applies an existing framework [5] to evaluate the suitability of using MAS to call management. Section 4 details the first phase of the development of MAS using a semi-formal requirement language *i** [22]. Section 5 concludes with a description of future work and anticipated challenges.

2 Call Management and Related Work

We propose to perform real-time monitoring of the CMC while RMs are performing their sales and to adjust the call flow rate to each RM according to specific criteria to be described in this section. To provide improved call routing and dynamic call flow control for both inbound and outbound calls, a distributed intelligent system will

provide assistance to RMs in serving their end customers (or potential customers) to ensure the best match between RMs and end customers. This section describes recent CMC-related research which deals simultaneously with both monitoring the performance of the RMs and matching them with end customers.

A CMC operation is complicated by the varying number and nature of products offered by its corporate clients. Much work has been done on customer relationship management and appropriate matching with customers based on RM performance and product knowledge. For example, in selling travel packages on behalf of a travel agency, a CMC would do well in matching end customers to well-informed RMs with appropriate knowledge about the destination and its traditions. A typical RM matching technique is segmenting customers into social and cultural segments according to their postcodes and surnames [21]. Supporting tools to create customer profiles exist, e.g. [14]. A corresponding RM profile may depend on the age, sex, culture, language proficiency, experience and product knowledge. Our proposed intelligent system will be used as a skill matcher between end customers and RMs based on their profiles. This makes RMs more convincing to the customer and increases the chance to achieve a sale. In targeting potential buyers with outbound calls, the system dials numbers automatically according to a customer target list previously loaded. The system retrieves the customer's details from the database, displays the details, and provides the RM with a script to use and guidelines to help in providing an adequate service to the end customer. For outbound calls, the proposed solution will create a specific calling target list for each RM and product based on his/her skills and profile.

RMs profiling is implemented by some companies before hiring. Psychometric tests are carried out for new employees during the hiring process to enable future matching with customers [9]. In their most basic versions, an interviewee takes a 10 min questionnaire which gets used to build a profile and a skill matrix. For example, outbound RMs need to be extroverts with an ability to generate excitement and handle rejection, while inbound RMs need the ability to listen and solve problems. Tools to profile employees during the initial staffing phase (e.g. call centre simulation [9]) are also often used. These will provide initial RM profiles for our system which will dynamically adjust according to RMs performance. The proposed system will assess human interactions and continually evaluate the RM's skills and match with an end customer as the sale/call progresses (in real time). It will recreate the RM's calling target lists based on the latest skill/profile evaluation.

For inbound calls, customers dial a number reaching the CMC which has its own private automatic branch exchange (PABX or PBX). A call routing and distribution routine that minimizes inbound call costs by reducing per call handling time is illustrated in [4]. A skill score is calculated based on the RMs previous call duration and profile. A score from 1 to 10 based on the likelihood to purchase the product is given to a customer according to some preloaded criteria. Customers with the highest scores are served first. Skill-based routing [16] directs calls to RMs based on skill levels and best match. The schedule of dialing end customers and the estimated call duration vary according to a RMs skill level and previous performance. For a call centre this would make a difference when predicting calls. The work in [17, 23]

attempts to predict the calls in a multi-skill environment. In another work [11], the schedule of calls is based on a skill matrix for the RM's skill-based routing, based on multiple priority skill levels. The Genesys system has a skill level for each RM used in call routing. The higher the skills level the more calls that the RM receives. In our proposed MAS system, this skill level will be automatically calculated by the agent system and matched to the skill level of the end customer. Variance in skill level can be equalized using collaboration.

Inbound customers can be directed to an interactive voice response unit [15] prompting them for options and may even ask for call reasons in a few words and then redirect the call to an automatic call distributor routing the call to the first available appropriate RM. Customers may hang up when they suffer from a long wait time [15]. Call centres that use toll-free services pay out-of-pocket for the time their customers spend waiting. This time can be reduced by providing customers with more automated services that serve them without the need to talk to a human RM thus saving the company a lot of expense and saving the customer's time and in some cases wasting a sales opportunity by customers hanging up and dropping their calls [15]. Call recording and automatic analysis for various cues on effectiveness of RM will be incorporated in our system (see Section 4).

3 Evaluating the Use of Multi-agent Systems for CM

Agents are highly autonomous, situated, and interactive software components. They autonomously sense their environment and respond accordingly. A multi-agent system (MAS) is a collection of interacting agents which are highly autonomous, situated, and interactive software components. They autonomously sense their environment and respond accordingly. Coordination and cooperation between agents that possess diverse knowledge and capabilities facilitate the achievement of global goals that cannot be otherwise achieved by a single agent working in isolation [5, 8, 10]. MASs have been shown to be highly appropriate for the engineering of open, distributed or heterogeneous systems. Deciding if a MAS approach should be used to solve a particular problem requires identifying its *suitability* and estimating its *relative cost* (versus alternative approaches) in terms of money, expertise, and time. To assess the suitability of MAS as an architecture for our call management system, we use a recently developed suitability framework [5]. The framework evaluates the applicability of a MAS solution to the particular problem (result is shown in Tables 1 and 2).

The framework has two steps. The first step identifies key features (or requirements), highlighting how appropriate a MAS solution might be in satisfying each of these features. The prominence of the features is also rated. If features rated as important (rating 4 or 5) are matched with a high level of appropriateness of a MAS solution (4 or 5), then a MAS solution is deemed highly suitable. For example, if the environment is dynamic and unpredictable, this is a strong indicator of MAS suitability as MASs are suitable for such environments. Applying the first step of the framework to our call management domain, the proposed solution is found to

Table 1 Feature ratings on the call centre domain

Feature	Appropriate (1–5)	Prevalence of the requirement in CMC	Importance (1–5)
Environment – open	3	The environment is open, there is no limitation on the number of end customers or usage profiles that can be created for both RMs and end customers	5
Environment – uncertain	3–4	There is no guarantee that RMs will match the end customer, some end customers might not have a matching profile so the closest match should be provided	5
Environment – dynamic	5	RMs change rapidly as the company has a high turnover rate. New end customer lists are provided by the main customer for the call centre to call on their behalf	4
Distributed – data	5	Data is distributed between a database and a calling on the dialler (e.g. Genesys system) which dials the numbers then connects to the platform providing a key to retrieve all end customers' information from the database, this is for outbound. For inbound the end customer calls in and then the data is retrieved from the database, if present, and the call is transferred to the RM	5
Distributed – resources	5	Resources are distributed and include client computers with application, client telephone, client application, client operating system	5
Distributed – tasks	5	Distributed tasks include sending e-mails, faxes and files to the main customers, receiving and making calls from/to the end customers. For the proposed solution there are distributed tasks like profile matching, profile analyser	5
Interactions – negotiation	5	There should be negotiation between the components to negotiate the best matching profile to route the call to	5
Reliability	5	The agent profile matcher should be reliable to accurately match profiles to facilitate sales and increase the conversion rate (number of sales made to number of calls)	4
Concurrency	4	Predict RM call ending, profile matching, profile analyser, and performance monitor agents will be working concurrently for more than one RM and end customer	4

Note: features with importance rating <4 and few others are omitted for space reasons

Table 2 Potential Agent roles, tasks importance, and appropriateness

Tasks	Task inputs	Task resources	Agency (1–5)	Importance (1–5)	Potential Agent roles
Monitor the RMs and keeps track of their service time patterns	Call outcome and duration	Call duration and outcomes	5	5	Performance monitor
Estimate call duration and the number of incoming calls	Average incoming calls per hour of day, number of RMs available	Call duration and available RMs	3	5	Load balancer
Transfer calls to appropriate RM according to the clients preferences and RM availability	Call start and end	Call routing to RMs	2	5	Router
Receive voice responses from end customers and routes calls based on their selection	End customer response	Workflow, End customer voice, played messages	2	3	IVR unit
Creates profiles for RM, end customers, and products	End customer, product, and RM details	RM details from HR, end customer, and product details from main customer	5	5	Profiler
This is the agent responsible for matching between product, end customer, and RM	End customer request and profile; RM's availability and profile; available products	RM, product and end customer profiles	5	5	Matcher

Note: tasks with importance less than three are omitted for space reasons

be operating a dynamic, distributed, open environment with software components operating remotely (see Table 1). These are characteristics (according to the framework) that suggest the suitability of a MAS. This is especially true given that there will be a lot of negotiation between the solution components, and moreover these components need to work independently and remotely which makes autonomous agents particularly appealing.

The second step in the framework focuses on the nature of the tasks required within the system and examines the potential suitability of agents for these tasks, Table 2. It examines the main tasks, performance measures, type of interaction between entities, task resources, and entities that execute the tasks. A rating is assigned according to the appropriateness of using agents (1–5) and importance of the task (1–5) based on this measure. Table 2 shows that all tasks with importance rating of 5 have potential agent attributes (agency measure) of 3 or more. This indicates that key system tasks can be decomposed in a way suitable for allocation to autonomous agents.

As indicated by the first step of the framework (Table 1), many requirements of the system point to the suitability of MAS for a call management system. This was confirmed with the second step of the framework which showed that many of the system tasks can be allocated to suitable agents requiring a degree of autonomy. In the next section, we proceed to undertake requirements analysis using stakeholder analysis technique with i^* which has been extensively used for MAS design [8].

4 Early Requirements Model for a Call Management MAS

The first phase of developing the CMC MAS is articulating the requirements in order to undertake an appropriate agent-oriented analysis. We perform requirement engineering (RE) activities informally with i^* [22], beginning with stakeholder requirement analysis and rationale for the new system. We use the i^* [22] modelling framework to represent MAS agents and the relationships between different agents. Our early requirement phase generates a high-level description of system goals and roles expressed in the i^* model. In a MAS, agents depend on each other to achieve goals and perform tasks. The resultant i^* model consists of two components: the strategic dependency (SD) model which models the different agents and the relations between them and the strategic rationale (SR) model which models the different tasks each agent has and the different proposed alternatives to accomplish these tasks. Other goal-oriented languages such as KAOS [7, 12] and AOR [20] could be used instead of i^* . However, previous experience [8, 18] has shown that the latter is a good language to express MAS requirements. In particular, the i^* ‘actor’ lends itself to readily model the actors and agents in a CMC. Our proposed system is composed of a number of Actors (Agents and Roles) (Fig. 1). The two human agents (the relationship manager (RM) and the end customer) are the main target of the proposed system which utilizes MAS to profile and match them in order to increase the likelihood of a sale commitment.

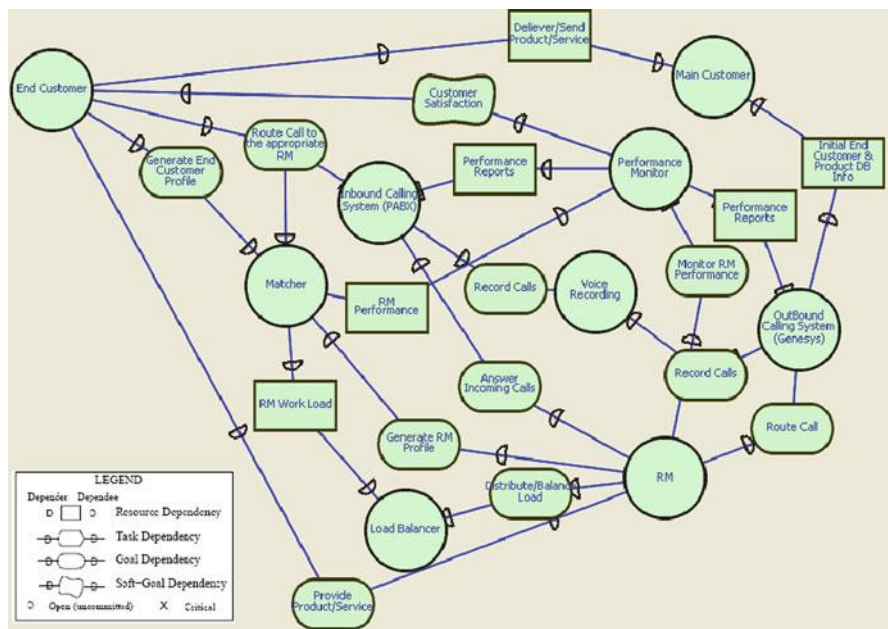


Fig. 1 SD diagram illustrating the dependencies between different actors of the system

The i^* model corresponding to our proposed solution identified nine agent roles (actors) as shown in Fig. 1. We have undertaken task and goal analyses for nine actors fully developing their dependencies. We illustrate the dependencies and tasks of matcher, end customer, performance monitor and RM. We illustrate their critical tasks and dependencies as they are the key actors in the call management system as described in Section 2. Due to space constraint and without loss of any critical insights, we do not detail the rest of actor dependencies in this section.

We identify nine different agent roles, developing the full i^* model showing the dependencies, tasks and goals for all roles identified. To illustrate the task identification with SR model of i^* , we zoom in on two key agent roles: *RM* and *performance monitor*, illustrating the dependencies, goals and tasks (shown in Fig. 2). Performance monitor and RM have a number of dependencies between them (e.g. voice recording shown in Fig. 1). These are fully analysed and tasks required to fulfil them are shown in Fig. 2. It turns out that performance monitoring and relations management involves at least 10 different tasks each.

For the *Performance Monitor* role they are as follows:

1. *Count calls made*
2. *Count sales made*
3. *Analyse performance:* This checks if there are any trends in the RM's call logging or performance, ex the RM logging all his calls as call backs or no sales.

RMs capabilities and performance. The system will offer RMs assistance in serving their end customers (or potential customers) to ensure the best match between RMs and end customers and to improve call routing. The system will lead to a higher rate of sales per call made/answered. Our research aims to provide a dynamic matching capability of the system that changes as products and end customers change. Given the distributed nature of the task, the dynamic nature of the data involved about products, end customers and RM themselves, we validate the use of multi-agent system technology to the system using a framework recently developed in [5].

Confirmation of the suitability of the software technology to the call management domain led us to the next step of requirement and stakeholder analysis using *i** modelling [22]. Our models in *i** were illustrated in this chapter. The next step in this research is to develop a prototype including key functionalities of monitoring and routing. Recent model-driven advances for multi-agent system development will be employed, e.g. [2, 3]. Using the call management expertise of the first author, a conceptual framework to allow testing and simulation of the domain will be undertaken.

References

1. F.G. Alessandro, J.P.d.L. Carlos and D.C. Donald (2004). Agents in object-oriented software engineering. *Software: Practice and Experience* 34(5): 489–521.
2. G. Beydoun, G. Low, B. Henderson-Sellers, H. Mouratidis, J.J.G. Sanz, J. Pavon and C. Gonzalez-Perez (2009). FAML: a generic metamodel for MAS development. *IEEE Transactions on Software Engineering* 35(4): 841–863.
3. G. Beydoun, G. Low, H. Mouratidis and B. Henderson-Sellers (2009). A security-aware metamodel for multi-agent systems (MAS). *Information and Software Technology* 51(5): 832–845.
4. F.J.B. Bogart, A.D. Flockhart, R.H. Foster, J.E. Kohler, E.P. Mathews and S.L. Skarzynski, (2000). *Optimizing Call-Center Performance by Using Predictive Data to Distribute Agents Among Calls*. United States, Avaya Technology Corp.: Miami Lakes, FL.
5. P. Bogg, G. Beydoun and G. Low (2008). When to use a multi-agent system? *11th Pacific Rim International Conference on Multi-Agents*, Prima 2008. Hanoi, Vietnam, Springer, Berlin. Vol 5357/2008: 98–108.
6. J.M. Brad Cleveland (1997). *Call Center Management on Fast Forward: Succeeding in Today's Dynamic Inbound Environment*. Annapolis, Md., Call Center Press.
7. J.M. Bradshaw, S. Dutfield, P. Benoit and J.D. Woolley (1997). KAoS: toward an industrial-strength open agent architecture. *Proceedings of the CIKM '95 Workshop on Intelligent Information Agents*. Baltimore, Maryland, USA: 375–418.
8. P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia and J. Mylopoulos (2004). Tropos: an agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Systems* 8(3): 203–236.
9. Doe, J. (2007). "Call Center Simulation." Retrieved 03/09/2008, 2008, from <http://contactcenter.limra.com/Products/Samples/Ccs.pdf>.
10. B. Fabio and R. Giovanni (2001). Developing multi-agent systems with a FIPA-compliant agent framework. *Software: Practice and Experience* 31(2): 103–128.
11. B. Gary, C. Plano, P.H. Lemon and McKinney (2000). Skills-based scheduling for telephone call centers. US Patent. USA. 6044355.

12. N. Hiroyuki, K. Takuya and H. Shinichi (2006). Analysis of multi-agent systems based on KAOS modeling. *Proceedings of the 28th International Conference on Software Engineering*. Shanghai, China, ACM: 926–929.
13. M.S. Jayashankar, F.S. Stephen and M.S. Norman (1998). Modeling supply chain dynamics: a multiagent approach. *Decision Sciences* 29(3): 607.
14. D.L. Larue, J.B. Ivey and T.M. Leonard, (1999). Generalized customer profile editor for call center services. US Patent. United States, MCI Communications Corporation (Washington, DC).
15. G.K. NoahGans, and A. Mandelbaum (2003). Telephone call centers:Tutorial,review and research prospects. *Manufacturing and Service Operations Management* 5: 79–141.
16. R. Thomas and D.J.a.P.D. Robbins (2006). Evaluating arrival rate uncertainty in call centers. *Simulation Conference, 2006. WSC 06. Proceedings of the Winter*. Monterey, California Winter Simulation Conference: 2180–2187.
17. T.S. Fisher, R.A . Jensen and M.I. Reiman (2003). System for automatically predicting call center agent work time in a multi-skilled agent environment US Patent. USA. US 6,553,114 B1.
18. Q. Tran, G. Beydoun, G. Low and C. Gonzalez-Perez (2008). Preliminary Validation of MOBMAS_ (Ontology-Centric Agent Oriented Methodology): design of a *Peer-to-Peer Information Sharing MAS*. *Lecture Notes in Computer Science* 4898: 73–89.
19. T.S. Vincent Conitzer (2007). AWESOME: a general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. *Machine Learning* 67 (1–2/may 2007): 23–43.
20. G. Wagner (2000). Agent-object-relationship modeling. In *Proceedings of Second International Symposium - From Agent Theory to Agent Implementation Together with EMCRS 2000*, Vienna, Austria.
21. R. Webber (2007). Using names to segment customers by cultural, ethnic or religious origin. *Journal of Direct, Data and Digital Marketing Practice* 8(3): 226–242.
22. E.S.K. Yu (1995). *Modelling Strategic Relationships for Process Reengineering*. Computer Science. Toronto, Canada, University of Toronto. Doctor of Philosophy: 131.
23. M.A. Zeynep Aksin and V. Mehrotra (2007). The modern call-center: a multi-disciplinary perspective on operations management research. *Production and Operations Management* 16(6): 665–689.

Design and Research on e-Business Platform Based on Agent

L.Z. Li and L.X. Li

Abstract The efficiency of enterprises can be improved and made more competitive by e-business. Consequently, e-business is developing in a swift and violent manner as a new type of business mode all over the world. But with the rapid increase of information on the Internet, the traditional technology cannot meet the requirement of information development well. Soon, high-efficient e-business system needs to be set up by a new kind of technology. Since the agent has the characteristic of movement, cooperation as well as some intelligence, it can compensate the shortcoming of the current e-business system. So how to lead the agent into e-business soon becomes the focus of academic and enterprises. This chapter analyzes the existing electronic business mode and designs a kind of the electronic business model based on the agent intelligence. It searches the goods information that match the customer request and negotiates goods price and the bargain conditions with seller, and recommends reasonable goods for the double win both the customer and the safer. The language of the system development is Java and use the B/S structure.

Keywords B2C e-business · Agent · JADE · Query · Negotiation

1 Introduction

e-business modes provide a framework for the system planning and design of the e-business. Most enterprise e-business activities are online transaction, business cooperation, and value exchange activities among customers, providers, and partners conducted electronically. Improving the efficiency of the transaction flow is the benefit goal that the enterprises pursue. The goal of e-business, however, is forming uniform business system service platform, realizing the integrated operation of the flows from material purchase to product sales and final to social service by

L.Li (✉)

Research Center of Cluster and Enterprise Development, Business School, Jiangxi University of Finance and Economic, China

conducting system integration and information fusion on MIS, VRM, SCM, CRM, and e-markets.

Agent is an emerging distributed computing technology which combines artificial intelligence with network technologies. The application of its intelligence, dynamic, and mobility into the e-business field can provide technical support for discovery of intelligent resources and automatic online transaction. The mobile agent is a kind of code or program which can move from a host to another on the network. In addition, it can choose when and where to move. During the moving, the status of agent is saved and encapsulated into information which is transmitted to a new host for continue execution. This plays an important role in improving the system delay and increasing the bandwidth utilization. Different from the traditional commercial operation, e-business is a new commercial operation mode which enables online shopping, online transaction, and online e-payment for enterprises, shops, and consumers via the Internet. It is realized mainly through EDI (electronic data interchange) and the Internet.

Aiming to design an agent technology-based e-business system framework model which supports such functions as query, transaction, cooperation, and auction, efficient online information query and gathering, effective information filtering to provide personal services for users, this document covers the following contents:

- (a) Introduces the concepts of e-business and agent;
- (b) Analyzes the agent-related theories and technologies, including the methods for developing agent-oriented software, negotiation mechanism among multiple agents, fault tolerance of the mobile agent, transition process of agent, communication between agents, and the JADE platform; and
- (c) Provides an agent-based original B2C e-business system.

This chapter is organized as Section 2 overviews e-business and agent technologies. Section 3 deals with JADE and we analyze and research these techniques, that are the basis of e-business development. Section 4 presents B2C e-business system model-based agent, Section 5 describes B2C e-business system design, and finally we come up with the solution to ensure the healthy, normal, and reliable operation of B2C e-business activities.

2 e-Business and Agent

In general, an additional channel of trade benefits consumers if it reduces their transaction costs and provides the product at a lower price than the existing channel. This phenomenon applies exactly to e-business. e-business has been rapidly growing in recent years. More and more people now purchase items online [1].

e-business refers to the business activities conducted electronically. To be more exact, e-business is the generic term of various commodity-exchange-centric activities in the modern society where technologies and economy are highly developed.

These activities are conducted by the people who master information technologies and business rules at a high efficiency and a low cost [2].

By redefining the traditional model of commodities interflow and reducing the sales media, e-business makes direct trading possible for producers and consumers and thus changes the way that the economy of the whole society operates to some extent. e-business makes the traditional business flow electronic and digital and features openness, interactivity, and globality [3]. Compared with the traditional transaction methods, e-business has many advantages. On one hand, it removes the time and spatial barriers. On the other hand, it provides rich information resources. Therefore, it makes the recombination of various society economy factors more possible and greatly influences the layout and structure of the social economy.

At present, many problems remain in the promotion of the e-business. To adapt to the new framework and order of the development of the e-business, it is necessary to deeply understand and research these problems to find the solutions to these problems.

- (a) Security of data transmission, that is, ensure that the data information transmitted on the Internet is not monitored and stolen by the third party. Bringing convenience to enterprises on one hand, Internet threatens the communication security of data on the other hand. Many data need to be transmitted during the e-business transaction and these data involve the trade secrets of the enterprises. At the present stage, one important barrier for the promotion of the e-business is security.
- (b) Completeness of data, that is, ensure that the data information transmitted on the Internet is not tampered. Despite many technical measures developed by the computer experts from various angles to protect the security of the e-business transactions, it remains difficult to ensure the security of transactions.
- (c) Identity authentication. During the e-business activities, two or multiple transaction parties need to exchange some sensitive information. In this case, it is necessary to confirm the identity of the other party.
- (d) Non-repudiation of transactions, that is, prevent the sender from denying sending the information and ensure there are evidences to verify the fact during transaction disputes. Many enterprises, both providers and purchasers have doubts about the many business activities on the Internet. Therefore, the execution and compensation of service contracts, fund safety, intellectual property protection, taxation, and other possible problems all hamper the development of B2B transactions.
- (e) Electronic payment and settlement technologies, that is, the way of electronic payment in the e-business. This problem is mainly caused by the reliability of the Internet and the speed of data transmission. At the present stage, some unreliable factors remain on the Internet, for example the unreliability of software, circuit, and system. The payments and settlements of electronic transactions require the cooperation of high-quality and high-efficiency electronic financial services. Today's infrastructure, such as servers, network cards, and buses, however, cannot keep up with the latest development in e-business.

In a word, the focus of the e-business research remains how to build a stable, safe, and reliable e-business system.

FIPA (Foundation for Intelligent Physical Agents), the largest agent standardization technology organization in the world, holds that agent is an entity residing in the environment and it can explain the data which is obtained from the environment and reflects the events occurred in the environment and can perform actions to affect the environment. Agent has such basic features as autonomy, pre-activeness, reactivity, learnability, and communication capability as well as such non-basic features as sociality, individuation, suitability, mobility, emotion, and personality [4].

3 JADE Technology

3.1 Agent Platform

A standard agent platform defined by FIPA is composed of the following two parts:

- (a) *AMS (agent management system)*: The agent supervises and manages the access and usage of the agent platform. An individual platform can have only one AMS. An AMS provides white and yellow page services and life cycle services [5]. It keeps an agent identifier directory (AID) and the agent status information. Each agent must be registered on the AMS to obtain an effect AID.
- (b) *DF (directory facilitator)*: The agent message transport system on the platform to provide default yellow page services, also called ACC (agent communication channel). It controls the exchange of all the messages within the platform, including the software which exchanges messages with the remote platform.

Figure 1 shows agent platform.
JADE refers to this standard architecture completely. Thus, when a JADE platform is started, an AMS and DF are automatically established and at the same

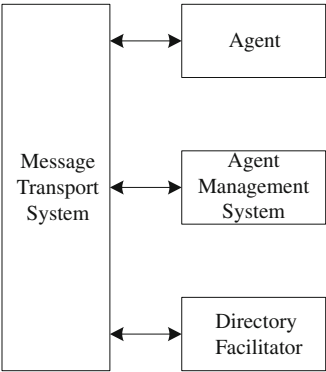


Fig. 1 Agent platform

time the ACC module allows message transmission. The agent platform can be established on multiple hosts with only one JAVE application and only one executed JAVA virtual machine on each host. Each JAVA virtual machine is equal to a basic agent container [6]. It provides an operating environment for the execution of agent and allows parallel execution of multiple agents on the same host. The primary container is the agent container where RMI is registered. It includes the AMS and DF. The other containers related to the primary container, however, provide a complete operating environment for the execution of any group of JADE agents.

3.2 Life Cycle of an Agent

According to the life cycle of an agent platform defined by FIPA, a JADE agent can be in one of the following states which are expressed by several constants in the agent class. These states are as follows:

- (a) INITIATED: In this state, an agent object has been established, but has not been registered by the AMS and thus has no name and address and cannot communicate with other agents.
- (b) ACTIVE: In this state, the agent object has been registered by the AMS and has a normal name and address as well as various JADE features.
- (c) SUSPENDED: In this state, the agent object is currently stopped, the internal threads are suspended, and no agent action is executed.
- (d) WAITING: In this state, the agent object is blocked to wait for other events. At the same time, the internal threads are in hibernation and will be woken up when the requirements are met (a typical case is that the message arrives).
- (e) DELETED: In this state, the agent object is dead. At the same time, the execution of the internal threads is terminated and the agent object no longer has registration information on the AMS.
- (f) TRANSIT: The agent object enters this state when it is transited to a new location. In this state, the system continues to cache the information which will be sent to the new location.
- (g) COPY: An internal state of JADE during the agent clone.
- (h) GONE: An internal stable state of JADE when the mobile agent is transited to a new location.

4 B2C e-Business System Model

By using the mobile agent technology, we design an agent-based e-business system. This system has the following features:

- (a) provides quicker and easier services for customers and shops;

- (b) saves the bandwidth;
- (c) increase the information retrieval efficiency;
- (d) provides intelligent transaction environment; and
- (e) realizes the coordination of multiple gents and the intelligent negotiation mechanism of e-business. In addition, this system has such functions as query, negotiation, auction, and personal services.

The B2C e-business system adopts the J2EE-platform technology-based B/S access mode and the multi-layer structure design idea. The structure of the model is as shown in Fig. 2. The system has the following layers:

Presentation layer realized by the JSP component in the web container, used to receive and analyze what the user entered and process it accordingly.

Control layer realized by the Servlet component in the web container. Based on what the user entered, perform corresponding control operations, including calling client agent and shop agent via Servlet with JSP.

Business logic layer mainly composed of seven classes of agents, including client agent, buy agent, shop agent, gatekeeper agent, seller agent, warehouse agent, and CIC agent. Among the seven classes of agents, client agent and shop agent are created by the buyer and seller and thus execute the tasks of both the buyer and the seller. Buyer agent is created by client agent or gatekeeper agent. It is the only mobile agent and execute the purchase tasks of the buyer. Gatekeeper agent, seller

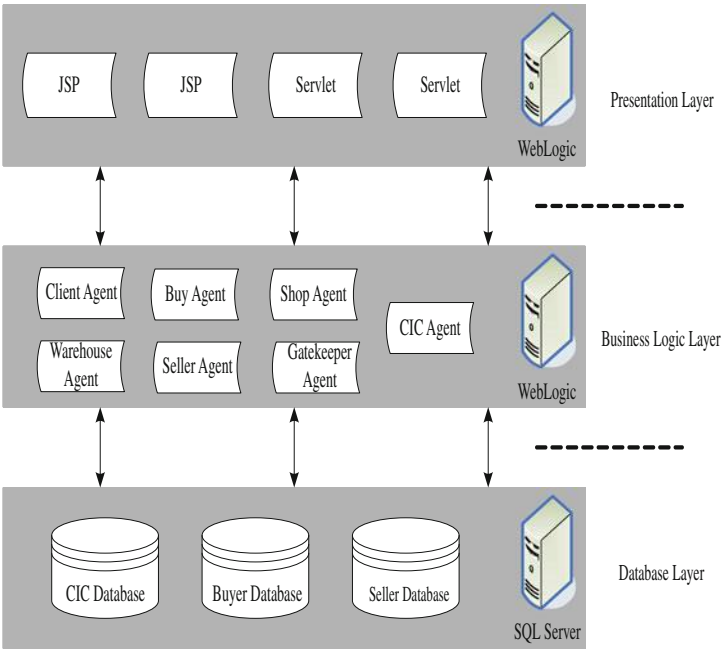


Fig. 2 Three-layer structure of the system

agent, and warehouse agent are created by shop agent. The main tasks of gatekeeper agent are authenticating the buyer agents from the buyer based on the strategies provided by shop agent. Warehouse agent stores the information about the goods of the seller. Seller agent negotiates with buyer agent on behalf of the seller. CIC (client information center) agent is the information center through which the buyer and the seller obtain the information of other users.

Data operate layer includes three databases, that is, the seller database storing the information about the goods of the seller, the strategies customized by the seller, and the knowledge base of the seller; the buyer database storing the strategies and knowledge base of the buyer; and the information center storing the basic information of both the buyer and the seller, including the information about the goods registered by the seller at other shops. These three databases interact with each other through the interfaces of the JDBC database.

5 System Design

This system is composed of three subsystems: CIC subsystem, seller subsystem, and buyer subsystem. It has such functions as search, transaction, negotiation, and auction as well as somewhat security and intelligence.

5.1 Buyer Subsystem

The buyer subsystem includes two kinds of agents: client agent and buyer agent. By logging into the buyer subsystem through the browser, the buyer can perform such operations as searching for and purchasing merchandises and modify/add/delete the strategies of the buyer. At the same time, the buyer subsystem can provide personal services for users and users can customize their own purchase strategies. Client agent is an intelligent agent. It can record the process and result of transactions, build its own knowledge base, and change the purchase model dynamically. For example, if client agent sends requests for dispatching buyer agent to a shop agent many times continuously with all the requests rejected, client agent will delete the ID of the shop agent from the directory at a proper time. Buyer agent is a mobile agent created by client agent [7]. It is dispatched to the seller subsystem to participate in transactions on behalf of the buyer.

5.2 Seller Subsystem

The seller subsystem includes four kinds of agents: shop agent, warehouse agent, gatekeeper agent, and seller agent. By logging in to the seller subsystem through the browser, the seller can perform such operations as merchandise registration, sales cancellation, and modify/add/delete the strategies of the seller. Shop agent is

an intelligent agent. It can record the process and result of transactions, build its own knowledge base, and change the sales model dynamically. For example, if a client agent sends requests for dispatching buyer agent to a shop agent continuously, the shop agent will make judgments intelligently. If decides the client agent is a malicious agent, the shop agent will record the client agent and reject the requests of the client agent or disable the IP address to which the client agent belongs.

5.3 CIC Subsystem

The CIC agent in the CIC subsystem is equal to a mediating agent. Each system has only one CIC agent. Its main function is storing and managing the information of the shop agents and client agents which participate in the system as well as providing query services to other agents. All shop agents and client agents who want to participate in transactions must register with the CIC agent [8]. The CIC agent will store the information in the CICDB. The CICDB has two functions: realize the registration of the client agent and the shop agent by storing the user’s ID and provide yellow page services by storing the information of all shop agents.

5.4 System Architecture

Based on the previous design, we use the JADE and J2EE technologies to design and implement a B2C e-business system which has such functions as query, shopping guide, transaction, coordination, and auction.

This system uses the Internet as the network environment and adopts the B/S structure (Fig. 3). In the B/S structured system, on the one hand, the user sends requests to many servers distributed in the network through the browser. On the other

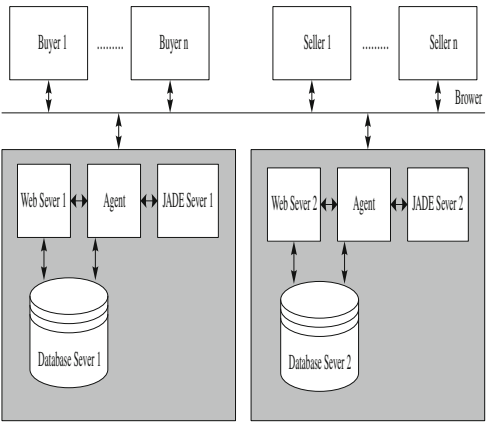


Fig. 3 System architecture

other hand, the server processes the requests from the browser and returns the information required by the user to the browser. By using the B/S structure, the client simplifies its work and needs to configure only few client software. The server, however, will be responsible for more work, such as access to the database and execution of the applications. The browser is responsible for sending requests. The web servers, however, have to be responsible for all the rest work, such as data request, data processing, result return, and generation of dynamic webpages.

This system is implemented by using Java as the development language. The mobile agent is a program distributed in various nodes in the network. It has executable codes and can be transmitted between the nodes. Tomcat is used as the web server and SQL server 2000 is used as the database.

6 Conclusions

Based on detailing the e-business and agent technology, this document designs an agent-based B2C e-business system model. This model has such functions as query, coordination, auction, and transaction as well somewhat adaptivity, intelligence, and mobility. It supports high-efficient online information query and gathering as well as effective information filtering and provides personal services for users and thus comply with the development trend of the third-generation e-business system.

References

1. Vidye Narayanan and Nicholas R. Jennings. An adaptive bilateral negotiation model for e-business setting. *Proceedings of the Seventh IEEE International Conference on E-business Technology (CEC'05)*, 2005.
2. D. Fensel. *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*. New York: Springer.
3. Asuncion Gomez-Perez, Mariano Fernandez-Lopez, Oscar Corcho. *Ontological Engineering: With Examples from the Areas of Knowledge Management, E-business and the Semantic Web*. London: Springer, 2004.
4. A. Chavez and P. Maes, Kasbah: An agent marketplace for buying and selling goods, *Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology*, London, UK 1996. 5(2): 75–90.
5. Peter R. Wurman, Michael P. Wellman and William E. Walsh, The Michigan internet AuctionBot: a configurable auction server for human and software agents, *ACM Autonomous Agents*, 1998. pp. 301–308.
6. Guttman, R., Moukas, A. and Maes, P. Agent-mediated electronic commerce: a survey, *The Knowledge Engineering Review*, 1998. 13(2): 147–159.
7. M. Wooldridge, *The Logical Modelling of Computational Multi-Agent Systems*, PhD Thesis Department of Computation, UMST, Manchester, 1992.
8. Piotr J Gmytrasiewicz, Matthew Summers and Dhruva Gopal. Toward automated evolution of agent communication language. In: *Proceedings of the 35th Hawaii International Conference on System Sciences*, HICSS-35, 2002.

Research of B2B e-Business Application and Development Technology Based on SOA

Li Liang Xian

Abstract Today, the B2B e-business systems in most enterprises usually have multiple heterogeneous and independent systems which are based on different platforms and operate in different functional departments. To deal with the increased services in future, an enterprise needs to expand its system continuously. This, however, will cause great inconvenience to the future system maintenance. To implement e-business successfully, a unified internal e-business integration environment must be established to integrate the internal system and thus realize a unified internal mechanism within the enterprise e-business system. The SOA (service-oriented architecture), however, can well meet the above requirements. The integration of SOA-based applications can reduce the dependency of different types of IT systems, reduce the cost of system maintenance and the complexity of the IT system operation, increase the flexibility of the system deployment, and at the same time exclude the barrier of service innovation. Research and application of SOA-based enterprise application systems has become a very important research project at present. Based on SOA, this document designs an enterprise e-business application model and realizes a flexible and expandable e-business platform.

Keywords B2B e-business · SOA · Web service · Architecture of system

1 Introduction

Promoted enthusiastically by merchants, the main body of e-business services, B2B e-business greatly promotes the efficiency of enterprises on supply, inventory, transportation, and flow of information. For a commercial enterprise in the circulation field, B2B e-business activities almost cover all its operations and management because it does not deal with production. Through B2B e-business, a commercial

L.L. Xian (✉)

Research Center of Cluster and Enterprise Development, Business School, Jiangxi University of Finance and Economics, Jiangxi 330013, China
e-mail: llx@jxufe.edu.cn

enterprise can make proper orders, reduce inventory, and promote sales through networks by timely obtaining the correct consumer information, thus improving the efficiency and reducing the cost [1].

With the popularity of B2B e-business, the requirements for cross-enterprise supply chain cooperation increase day by day. In this case, considering the requirements for the automation of business process, the subsystems should be integrated organically and interconnected on the business process, so that the interaction between the data and the applications of the sub-systems is reflected on the intercrossing of the functional modules [2]. At present, however, the functional subsystems of the e-business platform within the enterprise are independent systems, the application of the systems cannot meet the requirements for supply chain integration, and the departments are information isolated islands which are independent from each other. As a result, electronic information interaction and application interaction become the bottleneck. In this background, SOA-based e-business system comes into being. It adopts the SOA software design to resolve the above problem. By reorganizing different functions based on the business requirements, SOA design can change the organization of the system functions at will as required, just like pilling up building blocks. The SOA-based e-business platform is based on the internal supply chain. It requires that the departments within the enterprise interconnect with each other through LAN and connect to the outside in an open manner, so that a B2B e-business mode is implemented.

In this chapter we review some of the latest techniques that are being used in the line of SOA. The rest of the chapter is organized as Section 2 deals with working principles, features of SOA, Section 3 describes core technologies of SOA, Section 4 presents overall design of SOA-based B2B e-business platform, Section 5 presents system implementation process, and finally conclusions are drawn in Section 6.

2 B2B e-Business and SOA

Since the late 1990s, the attractiveness of using web sites as an integrated marketing communication media has increased as more users have accessed the Internet [3]. The advancement of the information technology (IT) has enabled a variety of products and services to be displayed without consideration of the physical aspects of space and time [4]. As a result, the “e-market,” a virtual place for online transactions, has become a center of attention and great efforts have been put into identifying the major success factors in e-business, the electronic commerce of business in the e-market [5].

SOA enable businesses to take advantage of connectivity and transaction processing to create a loosely coupled system integration, it supports the diversification of commercial relations and transaction processing, and it can put a different cross-regional commercial relations between enterprises and linking them to deal with the needs of business [6]. By using SOA, enterprises can reuse existing assets and purchase a solution to reduce business-to-business (B2B) application integration

development costs and implementation time, and no hard-and-fast to rewrite existing software system or restart the development. In short, SOA provides a reliable technology foundation for e-business between enterprises and has brought new opportunities for the development of enterprises [7].

The interoperability barrier of the B2B e-business applications makes it difficult for the enterprises to quickly find proper trading partners and trade with them [8]. At the same time, it slows down the development that enterprises participate in e-business. Different e-business follow different object models and adopt different communication protocols, which are different from each other both in data description and in business flow description. These differences make system interoperability difficult. This is because when any integration party changes its realization mechanism, the other party will have to make changes accordingly. Otherwise, the two parties may run the risk of coupling failure. In the future development of the e-business, the interoperability among various e-business applications will be an open question. SOA, however, proves to be the best solution [9].

SOA is a component model. It connects the different functional units (services) of the applications through the well-defined interface and agreements among these services. The interface is defined in a neutral manner. It is independent of the hardware platform, operating system, and programming language which are used to implement the services. In this way, the services in various systems can interact with each other in a unified and general manner. In SOA, resources are provided to the other members in the network, as independent services which can be accessed in a standard manner. Compared with the traditional system structure, SOA determines a more flexible loose coupling relation between the resources [10].

According to Gartner, SOA is a client/server software design method and an application composed of the software service provider and the software service user. The difference between SOA and most general client/server models is that SOA stresses the loose coupling of the software components and uses independent standard interface to final form service-oriented architecture integration.

SOA is a coarse-grained and loosely coupled service architecture. The services it provides communicate with each other through a simple and precisely defined interface without involving the bottom programming interface and the communication model. Such a model features loosely coupled, coarse-grained, and well-defined interfaces, message-based communication, and non-status service design.

3 Key SOA Technologies

3.1 Web Service Overview

As an important SOA implementation, web service is a relatively new technology widely accepted by people.

This is because web service provides a distributed computing method for the heterogeneous applications integrated in the Internet. It can integrate the applications

running on the distributed servers which are connected through Intranet, Extranet, and Internet together. The elements of web service are as follows:

3.1.1 XML Language

XML (eXtensible Markup Language) is a text-based markup language. It strictly defines portable structured data. It can be used as a language to define the data description language. In service description, XML is the mechanism of the basic data types and all service description technologies are presented with XML.

3.1.2 SOAP

SOAP (simple object access protocol) is a lightweight protocol used to exchange information in a decentralized and distributed environment. SOAP aims to allow clients and applications running on the Internet to exchange text information using standard methods. The SOAP standard has three main parts: SOAP envelop, encoding rules, and RPC presentation.

3.1.3 UDDI

UDDI (Universal Description Discovery and Integration) is a set of realization standards and rules for web service-oriented information registry center. The UDDI registry center is established to publish and discover web services.

3.1.4 WSDL

WSDL (Web Services Description Language) is a XML-based language which is used to describe the methods provided by web services and how to call these methods. The WSDL document describes service functions, the location of the services on the network, and some instructions about how to access the services.

3.2 Business Process Execution Language (BPEL)

BPEL (Business Process Execution Language) is a XML-based language used to describe the business process. Each step described by it is realized by web services. In other words, BPEL is a language which combines multiple web services in the point-to-point business process. According to the business process requirements, it organizes the current web services as a new web services which is the same as each single web service seen from the outside. The outside can call the BPEL process by calling web services.

3.3 Struts Structure

Struts is the implementation of a J2EE-based web MVC (model view controller) framework mode. The main component of Struts is a general control component

which provides entry point for processing all HTTP requests sent to Struts. The control component intercepts these requests and deliveries them to the corresponding action classes (these action classes are all subclasses of action class). In addition, the control component is also responsible for filling Form Bean with corresponding request parameters and transmitting Form Bean to the action class. By accessing Java Bean or calling EJB, the action class can implement the core business logic. Finally, the action class transfers the control right to the subsequent JSP files which then generate a view. All of these control logics are configured through the Struts-config.xml file.

Figure 1 shows function and theory of struts structure.

3.4 EJB

As a part of J2EE, EJB (Enterprise Java Bean) defines a standard which is used to develop component-based enterprise multiple utility programs. It has such features as network service support and software development kit (SDK). Actually, EJB is a component model technical specification defined by Sun Microsystems. An EJB component must comply with the regulations about related interface, implementation, and deployment description in the EJB component model specification. According to the specification, the EJB component must be compiled with JAVA language. It is used to present the reusable business logic components in the multi-layer distributed enterprise applications to meet special business requirements

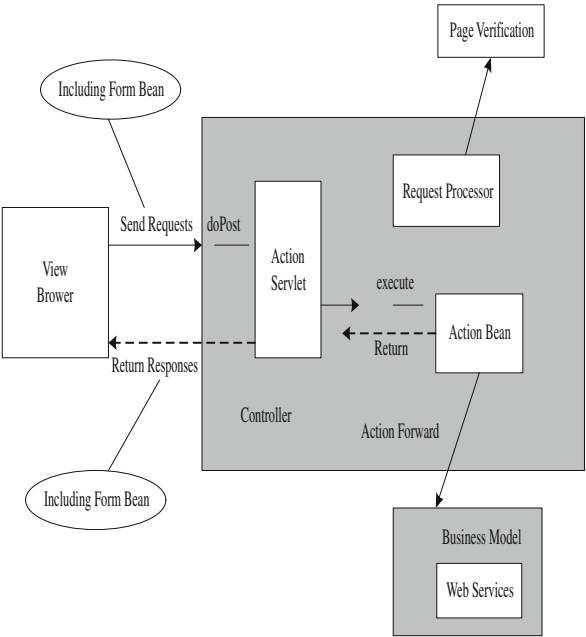


Fig. 1 Theory of struts structure and function

in the applications. The EJB component must be deployed in the EJB container to run. In addition, it can obtain various services provided by the EJB container, such as persistence, security, transaction control, and concurrency control. EJB components fall into three categories: session bean, entity bean, and message-driven bean.

3.5 ESB

ESB (enterprise service bus) is a pre-installed SOA implementation. It contains all basic function components required to realize SOA hierarchy. It is a flexible basic architecture used to integrate applications and services. Located at the center of SOA, ESB gives full scope to the advantage of SOA in software design by reducing interfaces in quantity, size, and complexity.

On higher hierarchies, ESB even provides such functions as service agent and protocol conversion and the application mode which we call ESB. As the service integration function provider of the SOA architecture, ESB has some common application modes, such as protocol conversion mode, message broadcast mode, and service matching mode.

The function of ESB is helping service integration, but not participating in business logics. Business logics should be encapsulated in business services or organized through business choreography services. ESB is only a mediator in SOA. Thus, the changes of ESB in future will not have great impact on the service requester and the service provider. This is because ESB is always transparent to the users.

4 Overall Design of SOA-Based B2B e-business Platform

Generally, an SOA-based B2B e-business platform system is composed of four parts: presentation layer, control layer, business (service) layer, and the data layer. During design, you can adjust the e-business platform system for specific services according to the features of enterprise applications and the requirements of the development platform and operating environment. Through user identification and with the help of the UDDI service registry center, the system automatically locates the required services according to the preset service combination method. Then, it binds and encapsulates the required services and finally realizes the users' operations. Objectively, the implementation, location, and transmission protocol of all the services at the business layer are transparent to the service caller. The foreground applications communicate with only the enterprise service bus. They send all business requests to the enterprise service bus. Without knowing whether the service exists at the business layer, the enterprise service bus locates the requested service at the business layer with the help of the UDDI service registry center. Then, it calls the service as required and returns the result to the foreground after service calling. In this way, SOA-based system deployment features are really implemented.

Considering the special requirement that the enterprise may establish system sharing with other enterprises in this industry, two UDDI service registry centers are set in the service resource library at the control layer. One of the UDDI service registry center is the local service registry center which is mainly used to register the services in the system to meet the service calling requirements in the system. The other is the foreign service registry center which is mainly used to call the services of other enterprises as required. Figure 2 shows architecture of system.

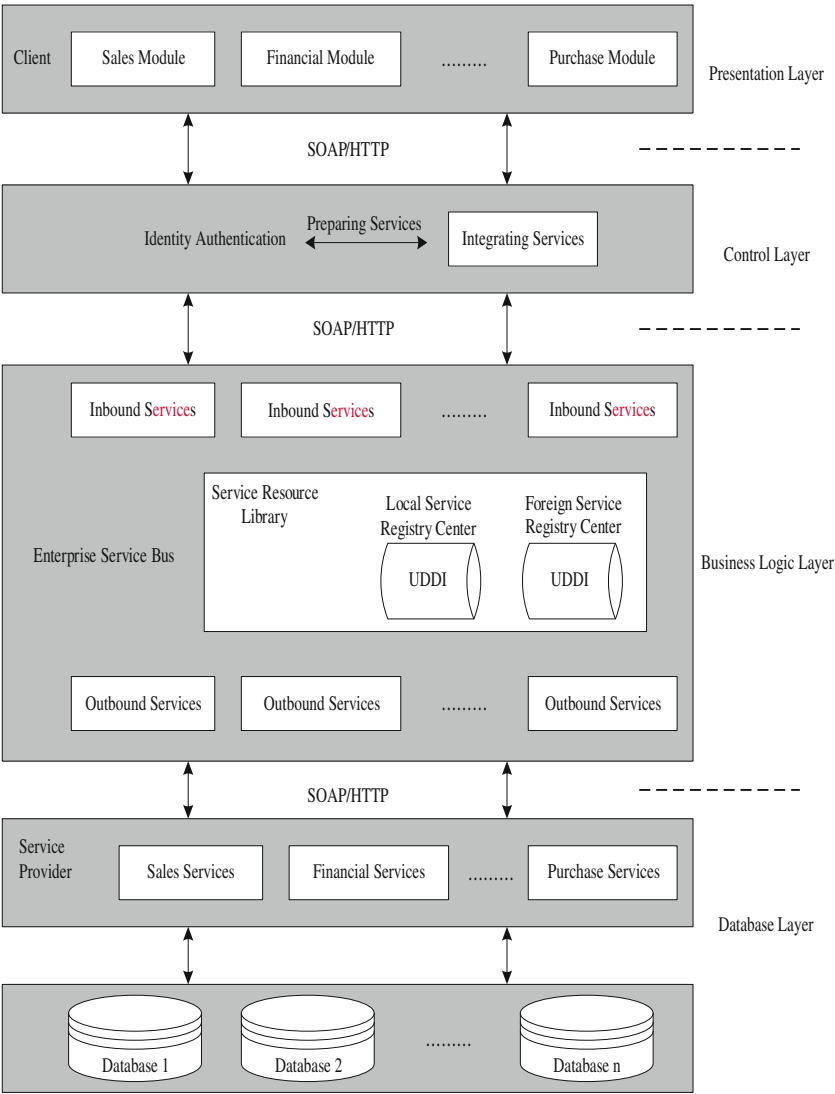


Fig. 2 Architecture of system

5 System Implementation Process

As shown in Fig. 3, the system mainly uses the MVC three-layer structure of Struts to implement the separation of the view layer and the business layer and uses the EJB Bean technology to implement the separation of the business logic layer and the data layer. This advanced development method will provide convenience for future system maintenance:

- (a) Analyze and design the database based on the business requirements and then perform encapsulations using the Entity Bean CMP technology and database.
- (b) Program the operation method in EJB Session Bean and opiate the database by calling Entity Bean.
- (c) Release the developed EJB Session Beans into web services and integrate the operation functions of all web services in the system implementation class ChemSoaMgr at the client.
- (d) At the control layer, call the method of the corresponding web service in the implementation class ChemSoaMgr, which will be used by the Action Bean based on the functions of the Action Bean.
- (e) Finally, at the control layer, ActionServlet calls the corresponding Action Bean based on the applications of the front-end view layer to implement the integration of the entire system.

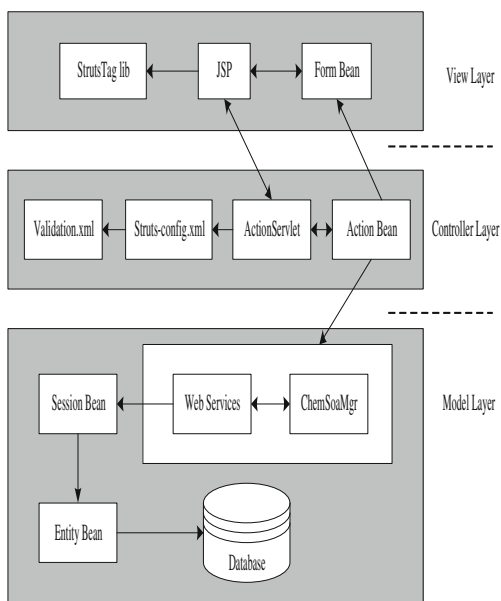


Fig. 3 Code structure of system

6 Conclusions

In general, this document implements the function requirements of the SOA-based B2B e-business platform. In addition, it well summarizes the development procedure of the SOA-based software system. This, however, will provide precious experience for future system improvement. Finally, this document shows the advantages of the SOA-based software systems, that is, flexible deployment, cross-platform, high system reusability, development time saving, and high expansibility. Thus, enterprises are really freed from solving technical problems to focus on providing excellent services.

References

1. E. E. Grandona and J. M. Pearson. Electronic commerce adoption: An empirical study of small and medium US businesses. *Information & Management*, 2001, 42: 197–216.
2. S. Leea and Y. Parkb. The classification and strategic management of services in e-commerce: Development of service taxonomy based on customer perception. *Expert Systems with Applications*, 2009.
3. J. Hayes and P. Finnegan. Assessing the of potential of E-business models: Towards a framework for assisting decision-makers. *European Journal of Operational Research*, 2005, 160(2): 365–379.
4. U. Lechner and J. Hummel. Business models and system architectures of virtual communities: From a sociological phenomenon to peer-to-peer architectures [J]. *International Journal of Electronic Commerce*, 2002, 3(6): 41–53.
4. D. Papakiriakopoulos, A. Poulymenakou, and G. Doukidis. Building E-business Models: An analytical framework and development guidelines[C]. In: *Proceedings of 14th Bled Electronic Commerce Conference*, Bled, Slovenia, June 25–26, 2001.
5. K. L. Taylor and Keefe C. M. Conlton. A service-oriented architecture for a health research data network. *Proceedings of the International Conference on Scientific, Stabstical DB Management*, Santorini Island, Greece, 2004, pp. 443–444.
7. M. Luo, B. Goldshlager, and L.-J. Zhang. Designing and implementing enterprise service bus (ESB) and SOA solutions. *IEEE International Conference on Services Computing (SCC'05)*, Vol.1. 21–28, 2005.
8. M. Huhns and M. Singh. Service-oriented computing: key concepts and principles, *IEEE Internet Computer*, 2005, 3(1):45–49.
9. C. Auer and M. Follack. Using action research for gaining competitive advantage out of the Internet's impact on existing business models. In: *Proceedings of the 15th Bled Electronic Commerce Conference eReality: Constructing the eEconomy*, Bled, Slovenia, Vol. 17–19(6), 2002, pp. 767–784.
10. C. Henry and R. S. Rosen-bloom. The role of the business model in capturing value from innovation: Evidence from Xerox Corporation's Technology spin off companies[J]. *Industrial and Corporate Change*, 2002, 11(3): 529–555.

Dynamic Inventory Management with Demand Information Updating

Jian Liu and Chunlin Luo

Abstract This chapter considers the dynamic inventory problem for a single product over a finite horizon and with periodic review. When stockout occurs, the customer may accept a substitute product. The demand can be observed and is assumed to be continuous with a probability density function of a known functional form, but with an unknown parameter. The inventory manager updates the knowledge of the unknown parameter by Bayesian rule and the observed value of demand. We show that the dynamic inventory problem with observed demand can be reduced to a sequence of single-period problem. Based on the result, we get the optimal order level of each period when the substitution probability is known. When the substitution probability is not known, we use the sufficient statistic to update the estimate of the substitution probability and get the similar result.

Keywords Dynamic inventory · Substitution · Unknown demand · Bayesian

1 Introduction

In most traditional inventory models, a common characteristic is the assumption that the demand distribution has known parameters and is static throughout the planning horizon. But in practice, it is frequently the case that the inventory manager is uncertain not only of the demand but also of the demand distribution. Scarf [9] pioneered the empirical Bayesian approach to this problem in which the inventory manager simultaneously optimally manages her inventory levels while learning about the demand distribution by observing demands over time. Azoury [1] developed conditions under which the dimensionality of the problem can be reduced: the solution of this problem can be obtained from the solution of a much simpler problem that is

J. Liu (✉)

School of Information Management, Jiangxi University of Finance & Economics, Nanchang 330013, China

e-mail: liujian3816@263.net

independent of the scale of the market. The retailer need only take the optimal stock level for the simple problem and scale it based on the current best estimate of the size of the market. Assuming the item is perishable, Harpaz and Lee [4] recognize that when lost sales are not observed, one should initially increase the inventory level to learn more about the demand distribution. Further assuming that demand has an exponential distribution with gamma prior on the mean, Lariviere and Porteus [5] derive a closed-form expression for the Bayesian optimal inventory level and confirm that it exceeds the Bayesian myopic inventory level in each period. Ding et al. [3] and Lu et al. [6] extend this “stock more” result to perishable inventory systems with a generous continuous demand distribution. But these research consider the single item case without any substitute, that is, when stockout occurs, the unmet demand may be backlogged or may become lost sales. But in practice, a customer may accept a substitute when the product is out of stock.

Many researchers have addressed joint inventory management for multiple products under customer substitution, without the added complexity of Bayesian learning. This work can be categorized into papers with inventory competition between multiple retailers (such as Parlar [8]) and papers that assume centralized control (such as Mahajan and van Ryzin [7]). Incorporating Bayesian learning, Chen and Plambeck [2] study the dynamic inventory management with customer substitution, but they only consider the discreet case where the demand is discreet. This chapter will follow the idea to study the continuous case with demand substitution under unknown demand distribution.

The rest of this chapter is organized as follows. Section 2 describes the basic model with constant substitution probability. Section 3 analyzes the above dynamic programming model and presents the algorithm to find the optimal policy. Section 4 deals with the case with unknown substitution probability. Concluding remarks are given in Section 5.

2 Basic Model

Consider a single item periodic review inventory problem with a finite planning horizon of N periods. At the beginning of period n ($n = 1, 2, \dots, N$), the inventory manager selects an inventory level y_n for the product. The time lag between ordering and delivery is assumed negligible; a nonzero lead time for ordering is not considered here. For each unit of the product, the purchasing cost is c and the selling pricing is p , with $p > c > 0$. Suppose the product is perishable, so at the end of each period, the inventory manager disposes of ending inventory with a salvage value q per unit, while the manager is charged a stockout penalty h per unit of shortage if the demand exceeds the inventory level. In addition, each customer who arrives when the product is out of stock is offered a substitute product. If a customer accepts the substitute, the manager receives a contribution margin of m per unit from selling the substitute. To avoid triviality, we assume that $p - c \geq m$, which guarantees that selling the product from inventory makes economic sense. In each

period when stockout occurs, we assume that each unit of excess demand generates an independent Bernoulli trial: each customer is willing to accept the substitute with probability r and becomes a lost sale with probability $1 - r$. The objective of the inventory manager is to maximize total discounted expected profit. We denote the discount factor by α ($0 \leq \alpha < 1$).

Assume that the demand at period n , X_n is observable (in most applications this is unrealistic because only sales are observed, but this model might apply when orders arrive through a call center or over the Internet and accurate records of orders are retained) and that the demand in different periods are independent and identically distributed. In each period, X_n is generated by a probability distribution with known density $f(\cdot|\theta)$ and unknown parameter θ with realization $\theta \in \Theta$. Given a prior distribution density $\pi_n(\theta)$ and a demand observation x_n , the posterior distribution density $\pi_{n+1}(\theta|x_n)$ is given by

$$\pi_{n+1}(\theta|x_n) = \frac{f(x_n|\theta)\pi_n(\theta)}{\int_{\Theta} f(x_n|\theta)\pi_n(\theta)d\theta} \quad (1)$$

When the demand x_n in period n is observed, the prior π_n is updated to the posterior π_{n+1} , and the posterior at period n becomes the prior for period $n + 1$.

Correspondingly, the sequence of marginal demand densities $\{g_n(x), n = 1, \dots, N\}$ satisfies

$$g_n(x) = \int_{\Theta} f(x|\theta)\pi_n(\theta|x_{n-1})d\theta \quad (2)$$

The marginal density is the Bayesian estimate of the demand distribution density in period n .

Denote the corresponding cumulative probability function by $G_{n+1}(x)$, that is

$$G_n(x) = \text{Prob}(X_n \leq x) = \int_{-\infty}^x g_n(x)dx \quad (3)$$

In this section, we consider the case that only demand parameter θ is updated and the substitution probability r is assumed known.

The expected profit for the n th single period $\Delta_n(\pi_n, y_n)$ is given by

$$\Delta_n(\pi_n, y_n) = pE \min(X_n, y_n) + m \cdot rE(X_n - y_n)^+ - cy_n + qE(y_n - X_n)^+ - hE(X_n - y_n)^+ \quad (4)$$

Let $v_n(\pi_n)$ denote the maximum total discounted expected profit over periods $n, \dots, N$ given that the demand density is π_n at the beginning of period n , then

$$v_n(\pi_n) = \max_{y_n \in R_+} \left\{ \Delta_n(\pi_n, y_n) + \alpha \cdot \int_{R_+} v_{i+1}(\pi_{i+1}(\cdot|x))g_n(x)dx \right\} \quad (5)$$

with

$$v_{N+1}(\pi_{N+1}) = 0$$

The Bayesian updating process described above results in a sequence of dependent demand distributions. Hence, the Bayesian dynamic program in (5) yields adaptive ordering policies that are dependent on history through the observation value x_n .

3 Algorithm to Find the Optimal Policy

We obtain the optimal inventory level y_n^* by solving the optimality equation (5). Because v_n depends on history only through the current state π_n , and π_n is updated by the observed demands which is independent of past actions, the above dynamic programming problem reduces to a sequence of single-period problems in which the demand distribution is the updated marginal demand distribution. Thus, finding y_n^* from Equation (5) is equivalent to solving the optimality equations

$$v_n(\pi_n) = \max_{y_n \in R_+} \Delta_n(\pi_n, y_n) \quad \text{for } n = 1, \dots, N, \quad (6)$$

which are linked only through the Bayesian information structure. For this version of the newsvendor problem, the optimal order quantity varies between decision epochs because the demand distribution is updated after observing the demand; however, the ordering policy has no effect on the state transitions.

A key feature of this problem is that the policy is myopic, that is, the future information does not affect the optimal order quantity.

Theorem 1: The optimal order level in the period n is given by

$$y_n^* = \overline{G}_n^{-1} \left(\frac{c - q}{p - m \cdot r - q + h} \right),$$

where $\overline{G}_n(x) = 1 - G_n(x)$.

Proof:

$$\begin{aligned} \Delta_n(\pi_n, y_n) &= pE \min(X_n, y_n) + m \cdot rE(X_n - y_n)^+ - cy_n + qE(y_n - X_n)^+ \\ &\quad - hE(X_n - y_n)^+ \\ &= p \cdot \left(\int_0^{y_n} xg_{n+1}(x)dx + \int_{y_n}^{+\infty} y_n g_{n+1}(x)dx \right) + m \cdot r \int_{y_n}^{+\infty} (x - y_n)g_{n+1}(x)dx - cy_n \\ &\quad + q \int_0^{y_n} (y_n - x)g_{n+1}(x)dx - h \int_{y_n}^{+\infty} (x - y_n)g_{n+1}(x)dx \\ \frac{\partial \Delta_n(\pi_n, y_n)}{\partial y_n} &= p \int_{y_n}^{+\infty} g_{n+1}(x)dx - m \cdot r \int_{y_n}^{+\infty} g_{n+1}(x)dx - c + q \int_0^{y_n} g_{n+1}(x)dx \\ &\quad + h \int_{y_n}^{+\infty} g_{n+1}(x)dx \end{aligned}$$

Setting $\partial \Delta_n(\pi_n, y_n) / \partial y_n = 0$, we get

$$\overline{G}_n(y_n^*) = \int_{y_n^*}^{+\infty} g_{n+1}(x)dx = \frac{c - q}{p - m \cdot r - q + h}$$

Because $\Delta_n(\pi_n, y_n)$ must attain its maximum value at some point and there is only one critical point,

$$y_n^* = \overline{G}_n^{-1} \left(\frac{c - q}{p - m \cdot r - q + h} \right)$$

is the optimal order level.

Therefore, the optimal policy of (5) is obtained by the following procedure:

- (1) after observing x_{n-1} , use Equation (1) to update $\pi_{n-1}(\theta|x_{n-2})$ to $\pi_n(\theta|x_{n-1})$;
- (2) calculate $G_{n+1}(x)$ by integrating Equation (3) over x ;
- (3) compute $y_n^* = \overline{G}_n^{-1}((c - q)/(p - m \cdot r - q + h))$; and
- (4) increment n by 1 and return to (1).

4 Updating Both Demand Parameter and Substitution Probability

In the above sections, we assume that the substitution probability is known; however, in some applications it may not be the case. Then the manager needs to update not only the demand parameter θ according to Bayes' rule, but also the substitution r .

Because all the demand can be observed, updating θ and r can be separated. We have discussed how to update θ in Section 2; in this section we focus on how to update r . At the beginning of the first period, the manager has a prior expected value, denoted by $\overline{r}_1$, for r . At the end of the first period, the manager gets the observation of the number of unmet demand ξ_1 and the probability r_1 . In general, at the beginning of $(n + 1)$ th period, the manager gets a sequence of observations of numbers of unmet demand $\xi_1, \dots, \xi_n$ and the probabilities $r_1, \dots, r_n$, then

$$\overline{r}_{n+1} = \left(\sum_{i=1}^n r_i \xi_i \right) / \left(\sum_{i=1}^n \xi_i \right) \quad (7)$$

is the sufficient statistic.

Similar to the case with known substitution probability, the policy is also myopic. To get the maximum total discounted expected profit, we only need to solve Equation (6) where r is replaced by $\overline{r}_n$, and the corresponding order level is

$$y_n^{**} = \overline{G}_n^{-1} \left(\frac{c - q}{p - m \cdot \overline{r}_n - q + h} \right)$$

Therefore, finding the optimal policy can be summarized as follows:

- (1) after observing x_{n-1} , $\xi_1, \dots, \xi_{n-1}$, $r_1, \dots, r_{n-1}$, use Equation (1) to update $\pi_{n-1}(\theta|x_{n-2})$ to $\pi_n(\theta|x_{n-1})$ and use Equation (7) to update $\overline{r}_{n-1}$ to $\overline{r}_n$;

- (2) calculate $G_{n+1}(x)$ by integrating Equation (2) over x ;
- (3) compute $y_n^{**} = \overline{G_n}^{-1}((c - q)/(p - m \cdot \overline{r}_n - q + h))$; and
- (4) increment n by 1 and return to (1).

5 Conclusions

An inventory manager should make ex ante ordering decisions based on future demand. However, when demand is unknown at the time of the decision, it must therefore be predicted. In this chapter, we have attempted to provide a Bayesian method for expectations formation, which helps the inventory manager to upgrade and sequentially accumulate knowledge with regard to the true but unknown parameter of demand. More specifically, this chapter considers the dynamic inventory problem with demand substitution under unknown demand distribution. We show that the dynamic inventory problem with observed demand can be reduced to a sequence of single-period problem. Based on the result, we get the optimal order level of each period when the substitution probability is known. When the substitution probability is not known, we use the sufficient statistic to update the estimate of the substitution probability and get the similar result.

Acknowledgment This work is supported by the National Natural Science Foundation of China (NO. 70861002) and the Science-Technology Project of Educational Department of Jiangxi Province (NO. GJJ09290).

References

1. Azoury, K. S. (1985) Bayes solution to dynamic inventory models under unknown demand distribution. *Management Science* 31:1150–1160.
2. Chen, L., and Plambeck, E. L. (2008) Dynamic inventory management with learning about the demand distribution and substitution probability. *Manufacturing & Service Operations Management* 10(2):236–256.
3. Ding, X., Puterman, M. L., and Bisi, A. (2002) The censored newsvendor and the optimal acquisition of information. *Operations Research* 50(3):517–527.
4. Harpaz, G., and Lee, W. Y. (1982) Learning, experimentation, and the optimal output decisions of a competitive firm. *Management Science* 28(6):589–603.
5. Lariviere, M. A., and Porteus, E. L. (1999) Stalking information: Bayesian inventory management with unobserved lost sales. *Management Science* 45(3):346–363.
6. Lu, X., Song, J. S., and Zhu, K. (2005) On “The censored newsvendor and the optimal acquisition of information”. *Operations Research* 53(6):1024–1026.
7. Mahajan, S., and van Ryzin, G. (2001) Stocking retail assortments under dynamic consumer substitution. *Operations Research* 49(3):334–351.
8. Parlar, M. (1988) Game theoretic analysis of the substitutable product inventory problem with random demands. *Naval Research Logistics* 35(3):397–409.
9. Scarf, H. E. (1959) Bayes solution of the statistical inventory problem. *Annals of Mathematical Statistics* 30:490–508.

Analysis of Market Opportunities for Chinese Private Express Delivery Industry

Changbing Jiang, Lijun Bai, and Xiaoqing Tong

Abstract China's express delivery market has become the arena in which each express enterprise struggles to chase due to the huge potential demand and high profitable prospects. So certain qualitative and quantitative forecast for the future changes of China's express delivery market will help enterprises understand various types of market conditions and social changes in demand and adjust business activities to enhance their competitiveness timely. The development of China's express delivery industry is first introduced in this chapter. Then the theoretical basis of the regression model is overviewed. We also predict the demand trends of China's express delivery market by using Pearson correlation analysis and regression analysis from qualitative and quantitative aspects, respectively. Finally, we draw some conclusions and recommendations for China's express delivery industry.

Keywords Express · Pearson correlation · Regression analysis · Market forecasting

1 Introduction

Express delivery industry has an important impact on China's economy in increasing the employment, enhancing the competitiveness of the export sector, and improving the technology industry's investment environment, being an indispensable industry for the economic development of China [1]. With China's accession to the WTO, the trend of economic globalization is more and more obvious and the global trade is increasing rapidly. The growth of trade is bound to lead to the prosperity of the express delivery industry. In addition, the "online shopping" has become consumers', especially the young families', main way to daily shopping, e-commerce express delivery will continue to maintain high economic growth. Both aspects will promote rapid development of express delivery industry. According to China

C. Jiang (✉)

College of Information Management, Zhejiang Gongshang University, Hangzhou, China
e-mail: johncabin@mail.zjgsu.edu.cn

Statistical Annual of 2007, China has more than one million express delivery employees with more than 500 billion yuan annual turnover. There are three major express delivery markets which are the international express, the domestic express, and the local town express [2], mainly distributed in the Yangtze River Delta Region (Shanghai, municipality), the Pearl River Delta Region (Guangzhou, provincial capital, and Shenzhen, special administrative region), and the Bohai Bay Region (Beijing, the capital of China).

Facing the dual pressures of the financial crisis and the international competition, the logistics industry's release on the planning of adjustment and revitalization at the right time will improve multifaceted problems of China's current express delivery enterprises such as many and miscellaneous, small and poor. Therefore, carrying on some demand forecasts of the development changes of China's express delivery market will help the express delivery industry to adjust and improve the business models timely, promoting their improvement and development.

2 Regression Prediction Method

2.1 Summary of Regression Model

Regression models are divided into linear regression model and nonlinear regression model. The nonlinear regression model began in the early 1960s, it developed faster in the 1980s, and has now become an important statistical discipline. In many practical problems, it is necessary to inspect the relationship between object y (response variable) and factor x (explanatory variable) which effects y . We can select a more reasonable model through analysis, observation of the scatter diagram, and correlative relation.

The best unbiased prediction was first proposed by Henderson in 1950 [3]. At that time, only the dependent variable's best unbiased prediction of the linear prediction function in the mixed linear model had been studied and later Royall and Kees Janvan Garderen studied the dependent variable's best unbiased prediction of the linear prediction function in generalized linear model, log-linear model, and general growth curve model. In recent decades, a lot of people studied prediction by using simple linear regression model [4, 5].

Royall and Pereira [6–8] studied the prediction of total in simple linear regression model by using linear regression theory, deriving the best linear unbiased prediction of total. There is a lot of research of prediction about the future observed values in the literature [9–12].

Leung and Daouk [13] constructed the generalized regression prediction model, making prediction analysis to economic time series. Christiaan et al. [14] made prediction to economic variables by using principal component regression (PCR) and principal covariate regression (PCovR). Syed Shahabuddin [15] analyzed the highly correlated relationship between the economic variables and the car sales by using multiple linear regression in SPSS.

2.2 Basic Theory for Regression Prediction Model

Regression prediction method is a regression equation on y set up about x , based on time variables x and observational variables or indicators for a dependent variable y . The basic idea is although the independent variables and dependent variables do not have a strictly certainty function, we may try to identify the most representative mathematical model of their approximate relationship that the regression equation, then calculate the required value prediction according to the regression equation. y and x in accordance with changes in the relationship between interdependence different, can be divided into:

2.2.1 Linear Regression Prediction

Linear regression analysis prediction method is a very broad approach to deal with the linear relationship between an independent variable and a dependent variable. It is simple to apply and can handle the economic statistical data with a causal relationship.

Regression model: linear: $y = a + bx$, where x is an independent variable or correlated variables; y is a dependent variable or predict variables; and a and b are regression coefficients.

Regression coefficients a and b are obtained based on least square method and the known sample data, the solving result is

$$b = \frac{\sum x_i y_i \times \frac{1}{n} \sum x_i \sum y_i}{\sum x_i^2 \times \frac{1}{n} (\sum x_i)^2} \quad (1)$$

$$a = \frac{\sum y_i - b \sum x_i}{n} \quad (2)$$

2.2.2 Curve Regression Prediction

A variety of regression methods are based on linear regression because many interdependence relationships between economic phenomena are not linear, but express as a variety of curve forms, which makes the process of predictive analysis more complex. It is necessary to study a simple method to describe the nonlinear relationship between complex phenomenon variables. The common method is linearizing the nonlinear function, calculating undetermined parameters, and then establishing a mathematical model to make predictions.

The method steps are first, draw scatter diagram based on the original data, and analysis that scatter diagram is similar with which typical function curves, to determine the function type between independent variable (x) and the dependent variable (y), such as exponential function and logarithmic function. Second, transform the determined function type to linear equation with mathematics deformation and calculate the undetermined parameters. Third, establish the mathematical model to make economic regression based on undetermined parameters.

Some common function types can be translated to the linear form of $y = a + bx$ through variable substitution. Several main curve models provided in SPSS are

Logarithmic model: $y = a + b \ln x$;
 Quadratic model: $y = a + b_1x + b_2x^2$;
 Cubic model: $y = a + b_1x + b_2x^2 + b_3x^3$; and
 Exponential model: $y = a e^{bx}$.

2.3 Prediction Model Test

Although the sample data after the establishment of a regression equation cannot be immediately used for the prediction of a practical problem, it must be used for various types of statistical test.

2.3.1 The Test of Goodness of Fit

Regression equation fit test is to test the sample data gathered in the samples around the regression straight line intensity, which determines the regression equation on the sample data representative of the degree. Regression equation with coefficient of determination R^2 general implementation. This indicator is based on the total square deviation and on the basis of decomposition, namely

$$R^2 = 1 - \frac{\sum (y - \hat{y})^2}{\sum (y - \bar{y})^2} \quad (3)$$

Observations usually fall on a regression line partly, that is, $0 < R^2 < 1$. It is said that linear regression fits better when R^2 is more close to 1; conversely, if R^2 and Vietnam close r to 0, the linear regression fits worse.

2.3.2 The Significant Test of Regression Equation

Regression equation significant test is the dependent variable with all independent variables of the linear relationship between whether a significant hypothesis testing. Regression equation of the general test of significance using F tests, namely, the average of the regression sum of squares with the average ratio of residual sum of squares.

$$F = (n - 2) \frac{\sum (\hat{y} - \bar{y})^2}{\sum (y - \hat{y})^2} \quad (4)$$

F statistic can reflect the fit of the regression equation, if the regression equation of the goodness of fit high, F statistic will be significant; F statistic more significantly, the regression equation of the goodness of fit is also higher [16].

3 Qualitative Analysis

The analysis can be made from demand and supply of the Chinese express delivery market development. It is necessary to establish an appropriate index system to evaluate demand, whose target must be able to directly or indirectly respond to demand for express delivery. We take an integrated economic indicator as an example of the demand for express delivery evaluation and analyze the relativity between GDP targets and express quantity.

Integrated economic indicators include GDP and the proportion of secondary industry and tertiary industry share. GDP refers to a country (region) of all resident within the framework of the territorial unit within a certain period of time by the production and delivery of value of finished products and service. With the growth of GDP, the demand for express delivery industry is growing and its growth rate is significantly higher than that of GDP. The industry structure in the primary industry (agriculture, forestry, animal husbandry, fisheries) has less demand for express delivery, so only the shares from secondary and tertiary industries are selected in the study. The relevant analysis between the GDP target and the demand for express delivery is carried out as follows: GDP's statistics is shown in Table 1 and the domestic demand for express delivery of the data is shown in Table 2 [17].

Table 1 GDP of China in 1991–2006 (unit: 100 million yuan)

Year	GDP	Year	GDP
1991	2,178.15	1999	8,967.71
1992	2,692.35	2000	9,921.46
1993	3,533.39	2001	10,965.52
1994	4,819.79	2002	12,033.27
1995	6,079.37	2003	13,582.28
1996	7,117.66	2004	15,987.83
1997	7,897.3	2005	18,386.79
1998	8,440.23	2006	21,087.1

Source: China Statistical Annual 2007

Table 2 Volume of express delivery of China in 1991–2006 (unit: 10,000 PCS)

Year	Volume of express delivery	Year	Volume of express delivery
1991	86.62	1999	1,572.58
1992	139.88	2000	2,097.26
1993	280.15	2001	2,721.75
1994	496.68	2002	3,483.5
1995	695.34	2003	4,820.49
1996	934.07	2004	6,590.63
1997	1,008.79	2005	8,818.95
1998	1,182.18	2006	11,944.17

Source: China Statistical Annual 2007

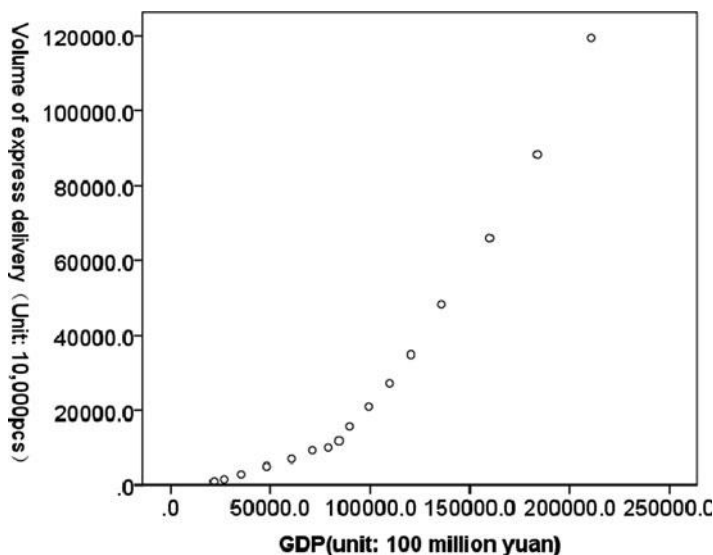


Fig. 1 GDP and the express delivery business scatter chart

The GDP of China from 1991 to 2006 and the express delivery business volume are illustrated in Fig. 1. We carry out the relevant examination by using the SPSS statistical analysis software to the express delivery demand and GDP, and according to the observation of the scatter diagram between GDP and the volume of express delivery business, the express delivery demand and the economic indicators showed a linear increase in the proportion of the trend. Using Pearson correlation to study the relationship between the ratios of data, we conclude that GDP and the volume of express delivery business with a correlation coefficient of 0.952, which means that GDP and the express business volume are highly related. Therefore, we can carry on the relevant forecast between GDP and the express delivery business volume. From 2001 to 2004, the express delivery business volume rose at 3% with GDP growth for each 1%. From 2006 to 2010, the Chinese GDP will maintain an annual growth rate of 8%. According to it, we can preliminarily estimate the business volume of express pace of development in more than 25%.

From the supply side, the main factors affecting the development of the express delivery market include the development of transportation network, in particular highways and aviation, investment, technical restraint, supply of talented people, and so on. For the local town delivery business, the primary factor affecting its development is urban road traffic indicators, including the year-end road pavement area and the per capital road pavement area. Local town express delivery industry in the city demands the time limit; therefore smooth traffic is an important guarantee to the time limit for express delivery.

4 Quantitative Analysis

The quantity of domestic express delivery business in 1991–2006 is shown in Table 3 and the curve of the domestic express delivery business in 1991–2006 is illustrated in Fig. 2.

Regression analysis is parameter estimation, testing the regression equation by using a sample, to predict or control the value of a variable. It will explore the correlated structure between variables, and especially describe the structure of causal relationship. The causal relationship between the two variables x and y can be described by the following model: $y = f(x) + u$.

Select the following four models to carry out the fitting to the express delivery volume curve (y is the express delivery volume, x for the time):

- Power model: $y = ax^b$;
- Exponential model: $y = ae^{bx}$;
- Quadratic model: $y = a + b_1x + b_2x^2$;
- Cubic model: $y = a + b_1x + b_2x^2 + b_3x^3$.

4.1 Postal Express Volume Forecasts

First of all, forecast the postal express delivery volume by applying regression analysis, as illustrated in Fig. 3.

According to the fitting analysis result of the SPSS statistical analysis software, the significant level of four models' F test is less than 0.0001, but in cubic models, $R^2 = 0.996$ and $F = 995.117$ are larger than that of other models; therefore

Table 3 The domestic volume of express delivery in 1991–2006 (unit: 10,000 PCS)

Year	Postal express delivery (EMS)	Nonpostal express delivery	Total
1991	56.67	29.95	86.62
1992	95.92	43.96	139.88
1993	215.62	64.53	280.15
1994	401.95	94.73	496.68
1995	556.27	139.07	695.34
1996	709.66	224.41	934.07
1997	687.89	320.9	1,008.79
1998	733.18	449	1,182.18
1999	909.13	663.45	1,572.58
2000	1,103.14	994.12	2,097.26
2001	1,265.27	1,456.48	2,721.75
2002	1,403.62	2,079.88	3,483.5
2003	1,723.78	3,096.71	4,820.49
2004	1,977.19	4,613.44	6,590.63
2005	2,288.03	6,530.92	8,818.95
2006	2,698.804	9,245.37	11,944.174

Source: China Statistical Annual 2007

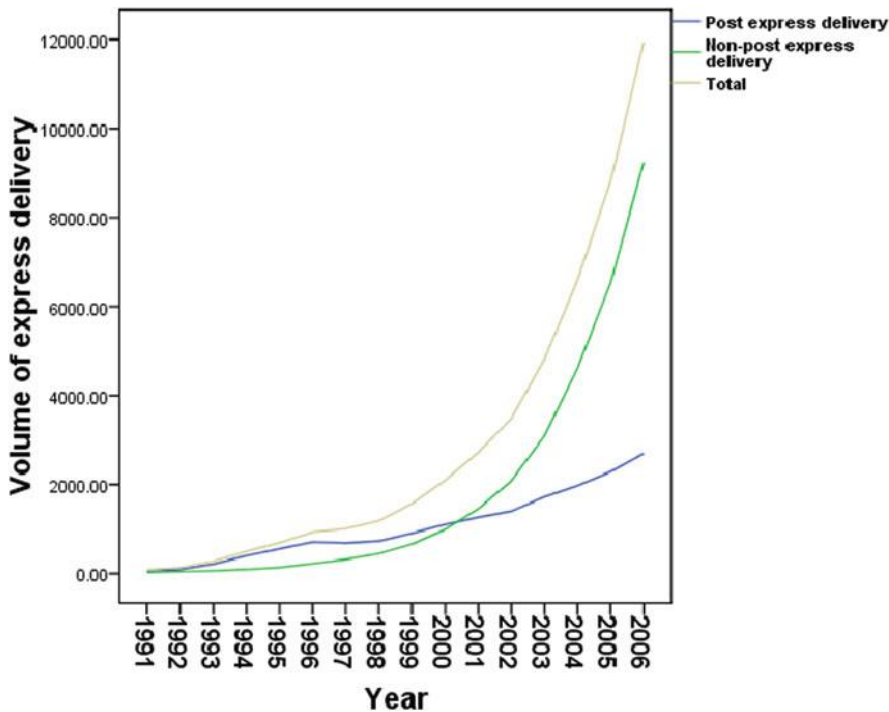


Fig. 2 Domestic volume of express delivery in 1991–2006



Fig. 3 The chart of postal express delivery volume



Fig. 4 The chart of cubic model

the cubic model is more reasonable. The fitting equation of the cubic model is $y = -1832.290 + 1999.440t - 176.109t^2 + 10.225t^3$. Carry out specific analysis of the cubic model, as illustrated in Fig. 4.

4.2 Nonpostal Express Volume Forecasts

First of all, forecast the nonpostal express delivery volume by applying regression analysis, as illustrated in Fig. 5.

According to the fitting analysis result of the SPSS statistical analysis software, the significant level of four models' F test is less than 0.0001, but in exponential models, $R^2 = 1.000$ and $F = 50,043.625$ are larger than that of other models; therefore the exponential model is more reasonable. The fitting equation of the exponential model is $y = 207.987e^{0.384t}$. Carry out specific analysis of the exponential model, as illustrated in Fig. 6.

According to the statistical results of the cubic model of postal and the exponential model of nonpostal, the volume forecast of the Chinese express delivery market in the next 10 years is shown in Table 4.

From our forecast data, in the next few years, the volume of postal express delivery business and that of nonpostal express delivery business will gradually increase, but the growth rate of nonpostal express delivery business will be higher than that of postal express delivery business. In the next 10 years, the average

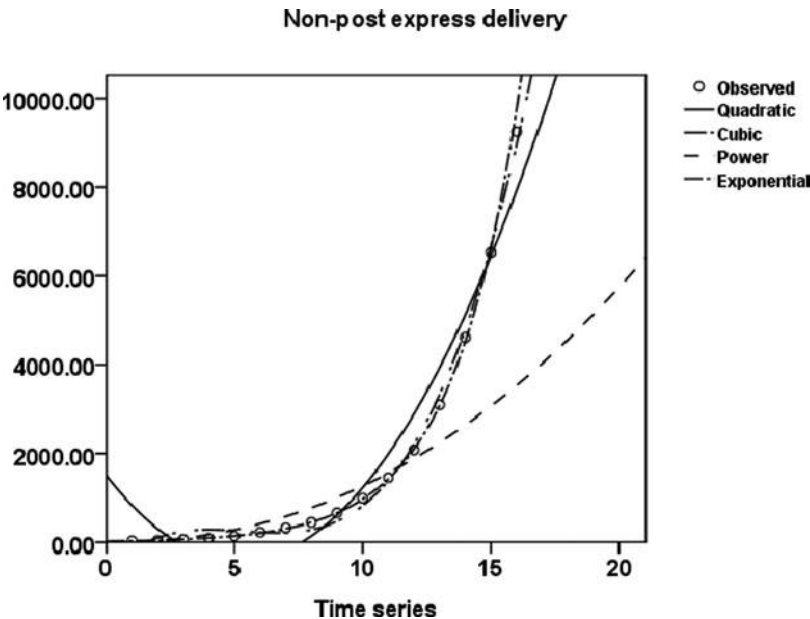


Fig. 5 The chart of nonpostal express delivery volume

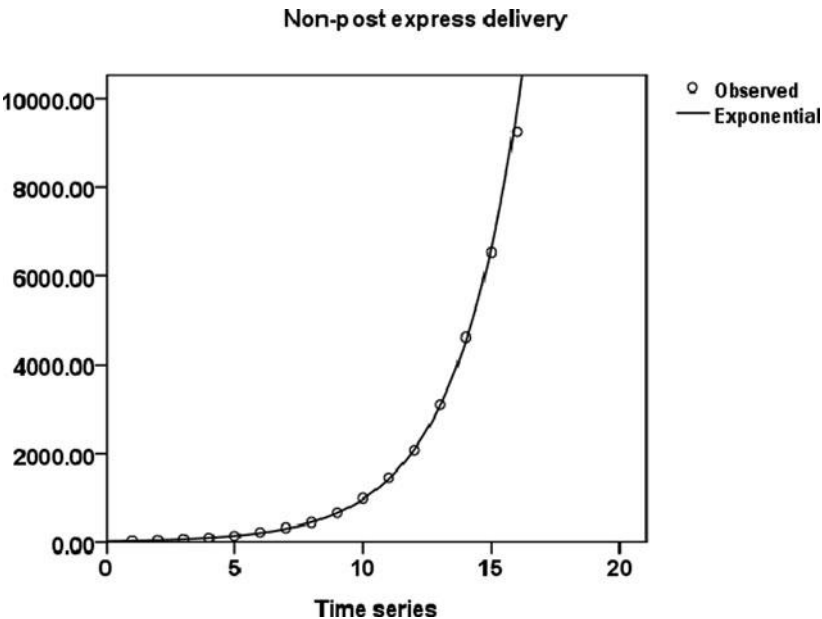


Fig. 6 The chart of exponential model

Table 4 The volume forecast of the Chinese express delivery in 2007–2016 (unit: 10,000 PCS)

Year	Forecast for postal express delivery	Forecast for nonpostal express delivery	Total
2007	31,498.1	142,269.1	173,767.2
2008	36,730.5	208,871.7	245,602.2
2009	42,715.0	306,654.1	349,369.1
2010	49,512.9	450,212.8	499,725.7
2011	57,185.6	660,977.8	718,163.4
2012	65,794.4	970,411.6	1,036,206.0
2013	75,400.7	1,424,705.3	1,500,106.1
2014	86,065.9	2,091,674.7	2,177,740.6
2015	97,851.2	3,070,882.6	3,168,733.8
2016	110,818.1	4,508,502.3	4,619,320.4

Table 5 The forecast of the annual growth rate of express delivery volume in 2007–2016

Year	Forecast of growth rate for nonpostal express delivery (%)	Forecast of growth rate for postal express delivery (%)
2007–2008	46.81	16.61
2008–2009	46.81	16.29
2009–2010	46.81	15.91
2010–2011	46.81	15.50
2011–2012	46.81	15.05
2012–2013	46.81	14.60
2013–2014	46.81	14.14
2014–2015	46.81	13.69
2015–2016	46.81	13.25

annual growth rate of nonpostal express delivery business will be 46.81%, which is much higher than that of the postal express delivery business (15.01%), as shown in Table 5.

5 Conclusions

As the express delivery market develops and tends to mature, the competition for its share has become increasingly fierce. Constant improvement of the express delivery industry's quality of service and increase of value-added services will drive enterprises' e-commerce trade to grow at a high speed, which has also brought broader market space.

Express growth and GDP growth are closely related. Logistics increase by two percentage points and express delivery increases by three percentage points, while the Chinese GDP increases by each one percentage. Together with such a high speed of development and the localization advantage, private enterprises will win the space for development. Insiders believe that the Chinese express delivery market will not be saturated until 20 years later. Although overseas tycoons try to seize the Chinese

market, relying on their advantages of funding, brands, and operating mode, their localization process needs time, and this is the opportunity for the Chinese express delivery industry to integrate and develop itself. Small- and medium-sized express delivery business' integration will be the trend in the next 2 years.

From the Chinese express delivery market demand for the regression analysis, we can see that the growth potential of nonpostal express delivery companies, which is mainly to the private, is considerable. At the same time, due to private enterprises accounting for express delivery market share of 75%, the average annual growth rate of nonpostal express delivery business will be higher than that of the postal express delivery business by three times over the next 10 years.

With current rapid development of the Chinese logistics industry, if the Chinese private express delivery companies can make full use of their advantages and overcome their own limitations, they will certainly play an important role in the field of logistics in the future.

Acknowledgments It is a project supported by philosophy and social science in Zhejiang Province (07CGLJ018YBX), the results of Center for Research in Modern Business, Zhejiang Gongshang University (the important research base for high school social and science of High Education Department), and the normal project of philosophy and social science in Hangzhou (D07GL07).

References

1. Li Qian, Lv Li Ping, and Yan Jing Dong (2008) China's express industry industry analysis. *Business Economics* 8: 110–111.
2. Chen Shi Yang (2007) Express industry in our country the status quo and countermeasures. *China Water Transport* 9: 194–195.
3. Henderson C. R. (1950) Estimation of genetic parameters. *The Annals of Mathematical Statistics* 21: 309–310.
4. Rao C. R. (1976) Estimation of parameters in a linear model. *The Annals of Statistics* 4: 1023–1037.
5. Kees Jan van Garderen (2001) Optimal prediction in loglinear models. *Journal of Econometrics* 104: 119–140.
6. Royall R. M. (1970) On finite population sampling theory under certain linear regression models. *Biometrika* 57: 377–387.
7. Royall R. M. and Herson J. H. (1973) Robust estimation in finite populations. *American Statistical Association* 68: 880–889.
8. Pereira C. A. B., and Rodrigues J. (1983) Robust linear prediction in finite populations. *International Statistical Review* 51: 293–300.
9. Yoshikazu Takada (1981) Relation of the best invariant prediction and the best unbiased prediction location and scale families. *The Annals of Statistics* 9: 917–921.
10. Gauri Sankar Datta, and Malay Ghosh (1991) Bayesian prediction in linear models: Applications to small area estimation. *The Annals of Statistics* 19: 1748–1770.
11. Richard M. Royall, and Dany Pfeiffermann (1982) Balanced samples and robust Bayesian inference in finite population sampling. *Biometrika* 69: 401–409.
12. C. Radhakrishna Rao (1987) Prediction of future observations in growth curve models. *Statistical Science* 4: 434–471.
13. Leung M. T., Chen A. S., and Daouk H. (2000) Forecasting exchange rates using general regression neural networks. *Computers & Operations Research* 27: 1093–1110.

14. Christiaan Heij, Patrick J. F. Groenen, and Dick van Dijk (2007) Forecast comparison of principal component regression and principal covariate regression. *Computational Statistics & Data Analysis* 51: 3612–3625.
15. Syed Shahabuddin (2009) Forecasting automobile sales. *Management Research News* 7: 32–42.
16. Yu Jian Ying, and He Xu Hong (2006) *SPSS Statistical Data Analysis and Application*, pp. 194–196. Beijing: Posts & Telecommunications Press (in Chinese).
17. National Bureau of Statistics of China (2008) *China Statistical Yearbook 2007*, pp. 210–252. Beijing: China Statistics Press (in Chinese).

Part VI
Development of Information Systems for
Creativity and Innovation

Explaining Change Paths of Systems and Software Development Practices

Kari Smolander, Even Åby Larsen, and Tero Päivärinta

Abstract This chapter discusses how systems development practices are shaped. Based on interviews conducted in ten development organizations and previous literature, we identify eight types of change paths in systems development practices: emergence, adoption, idealization, formalization, abandonment, informalization, entropy, and disobedience. We argue that the eight change path types provide an integrated theoretical framework on the study of how systems development practices change in organizations, projects, and among individual developers in a given context. We discuss how this framework complements existing theories and concepts of the contemporary literature on systems development.

Keywords Information systems development · Software development · Practice · Practice change

1 Introduction

Systems development practices have been studied over years by many IS scholars. However, no systematic research tradition on development process innovations had emerged [11]. Moreover, previous studies have been limited with regard to number of adoption decisions studied, types of innovations covered, or types of factors included in the study [11]. Studies of development practices have covered varying units of analysis: individual professionals, development projects, and development organizations without an integrated view [9].

We posed the research question: How do systems development practices change in development organizations? We analyzed changes in development practices in ten systems and software development organizations. The analysis is based on the NIPO

K. Smolander (✉)

Department of Information Technology, Lappeenranta University of Technology, P.O. Box 20, 53851, Lappeenranta, Finland
e-mail: kari.smolander@lut.fi

grid [9] which distinguishes between the intention vs. actual enactment of practices at the analytical levels of nondefined practices, individual practices, project-specific practices, and organization-wide practices. We identified eight stereotypical change paths of systems development practices in our target organizations: emergence, adoption, idealization, formalization, abandonment, informalization, entropy, and disobedience. We argue that the resulting framework of the change path types and the NIPO grid can be used to integrate and complete previous theories used for analyzing change in ISD practices in organizations.

After providing a conceptual and theoretical background for our study in Section 2, Section 3 declares the research process in more detail. The empirical grounding for the change paths for systems development is illustrated in Sections 4 and 5. Section 6 discusses previous concepts and theories of change in ISD practices in light of our results. Section 7 concludes with suggestions for further research.

2 Conceptual and Theoretical Background

In this chapter we focus on *system development practices* in context of systems and software development organizations. We chose to speak of practices (instead of methods or methodologies, for example), as it is well known that actual systems development work is shaped “in action” by adoption and adaption of individual practices rather than complete and philosophically consistent system development approaches and methods, e.g., [3, 14]. A *practice* is an ISD activity conducted repeatedly in a somewhat similar manner in systems development. This is consistent with common dictionary definitions of a practice, such as “something people do regularly” [2]. A practice can be explicitly defined, e.g., as a technique specified by a method, or emerge as a habitual response to a recurrent situation. An ISD practice existing in context of a development organization is an organizational practice, which has been defined as “the organization’s routine use of knowledge,” which “often has a tacit component, embedded partly in individual skills and partly in collaborative social arrangements” [7, 12, 19].

There exist different viewpoints on how organizational practices take shape. For example, Szulanski [19] argues that a “best practice” represents organizational knowledge, which can be transferred between a source and a recipient unit inside an organization as a replication of an organizational routine. On the other hand, Kostova and Roth [8] suggest that an organizational practice “evolves over time under the influence of the organization’s history, people, interests, and actions” and that it comes “to reflect the shared knowledge of the organization and tend[s] to be accepted and approved by the organizational members.” That is, a practice may be either rationally adopted or emergent in an evolutionary manner.

Any existing description of a practice implies that at least one stakeholder in the organization has intended that the description should be followed. However, ethnographical studies since the 1980s have shown that such canonical work practices often deviate from the actual actions taken [1]. Moreover, systems development organizations may, to an extent, follow undocumented practices [9]. In line with this, Pentland and Feldman [15] make a distinction between the performative and

Fig. 1 The NIPO grid – intention vs. enactment [9]

Scope of enacted practice	Intended scope of defined practice			
	N	I	P	O
None				
Individual				
Project				
Organization				

ostensive aspects of organizational routines. The performative aspect is “the specific actions taken by specific people at specific times when they are engaged in what they think of as an organizational routine.” The ostensive aspect is “the abstract or generalized pattern of the routine.”

To summarize the discussion above, we make an analytical distinction between the *intended scope* of a practice in question vs. its actual *enactment*, i.e., the extent to which a practice is actually followed at the moment. The NIPO grid [9] visualizes this distinction. The vertical dimension represents the actual enactment of a practice, which may be followed throughout the Organization, in a Project, by an Individual, or Not at all. The horizontal dimension represents the intended scope of the practice. Any practice discovered in an organization can be plotted on the grid, e.g., a practice intended to be used by all project members, but only enacted by some would be plotted in the “P” column of the second row. Practices which are plotted on the main diagonal (gray in Fig. 1) are enacted as intended, while practices plotted in the upper right half are not enacted as much as intended, and those in the lower left half are enacted more than intended.

3 Research Process

This study represents the second phase of an ongoing project focusing on the issues that shape systems development practices in organizations. We wanted to find an analytical concept to discuss what practices exist (and do not exist) in the target organizations, and then how they change and why. The empirical results are, in total, based on 45 interviews with developers and managers in ten IS development organizations (Table 1). The interviews were tape-recorded and transcribed.

The data analysis in this chapter focused on the developers’ own descriptions of how systems development tasks have been formed or changed in the target organizations. In the previous phase of the research, the NIPO grid was constructed, while we now recognized histories from the data explaining change in development practices. We interpreted the mentions of interviewees in light of the NIPO grid and categorized them accordingly. The resulting theoretical concept of the change patterns of development practices in light of the NIPO grid can be regarded as a synthesis of the insight we gained in the studied cases. This chapter presents the first results of our analysis in light of the NIPO grid, forming the stereotypical concepts. More data collection and analysis is needed for confirming the “theoretical saturation” of the identified change processes.

Table 1 Case organizations

Org	Employees	Business	Interviews
A	6 (Norway, one site)	Small web-based e-business solutions	1
B	8 (Norway, one site) + 10 (India, one site)	Logistics solution for electricity networks	2
C	42 (Norway, one site)	System development and administration	2
D	25 (Norway, one site) , (Serbia, one site)	Systems development and consulting	8
E	4 (local office), 184 (Norway altogether)	Systems development and consulting	2
F	800 (in multiple countries)	Operative systems for industry	7
G	100 (Finland, one site)	Automation systems for industry	8
H	200 (Russia, one site)	Software development services, subcontracting	5
I	60 (Finland, one site)	Systems for a specific kind of resource planning	5
J	80 (Finland, one site)	Internal systems development organization	5

4 Change Paths of ISD Practices

In the analysis, we identified contemporary development practices in each organization. Every case was quite different from each other. For example, a small logistics solution provider (Case B) had adopted iterative UI development and technical specification practices at the organization level. The biggest organization F, the operative systems provider, considered the “project model” defining project management and related practices as its key practice that was defined and enacted at the organizational level.

We drew a NIPO chart of each case and placed the identified practices in appropriate locations. We continued the analysis of the interview data and identified conditions that either enforce or erode the enactment of the identified practices. In each case, we identified 5–20 of such conditions. However, there were too many different practices and conditions to draw meaningful conclusions. Figure 2 shows an example of the analysis at this stage. In case F, two identified sets of key practices were mentioned, “Project model” and “ISD method definition.” Eleven factors, identified by the informants, caused pressures on intentions or enactment regarding these practices. For example, role diversification, project size and complexity, and outsourcing seemed to facilitate intentions toward defined organization-wide practices, whereas need to fit to the customer, missing method skills, nondefinable expert work, and negative attitudes could hinder the actual enactment of the ISD method in question.

Hence, we continued the analysis and decided to concentrate on the question whether we could be able to identify general patterns of changes in practices over

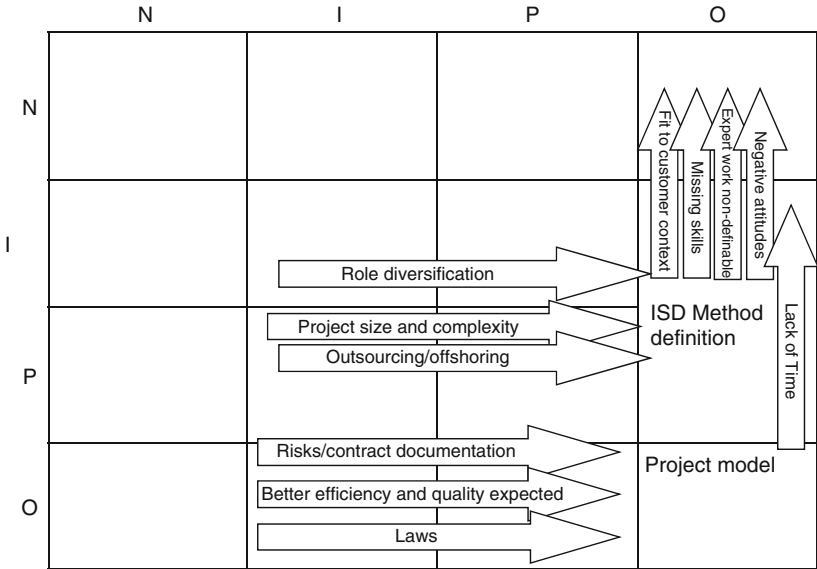


Fig. 2 Sample NIPO chart with practices and pressures

time using the NIPO coordinates. We carefully went through the data and analyzed pressures against the intentions and enactments of practices. We consider this analysis phase as selective coding [18], because we were no longer looking for categories and connections between them. Instead, the analysis required us to move back and forth between the data and theoretical constructs and most of the analysis happened at theoretical level, which we were constantly able to ground with the observations from the data. The core category emerged as “the change paths of ISD practices” and this meant in the analysis that we were able to describe the change in light of the theoretical constructs from the NIPO grid and simultaneously find evidence of such change paths of practices from the cases.

5 Eight Stereotypical Change Paths of ISD Practices

In the following, we will present the identified change paths (Figs. 3, 4, 5, and 6). These paths are stereotypical by nature. This means that these change paths are very general and certainly oversimplified for a specific change situation in practice. The empirical data provided examples of each of these change paths.

In *Emergence* (Fig. 3), a practice is enacted more widely than intended. This could happen when, for example, individuals or projects copy successful practices from other individuals or projects. This may happen over time and we found several examples of *Emergence* from the data. A middle-sized automation system

Fig. 3 Emergence and adoption

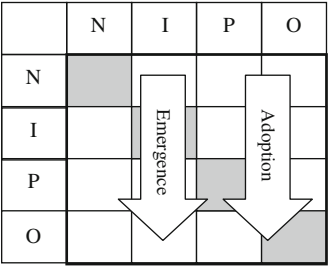


Fig. 4 Idealization and formalization

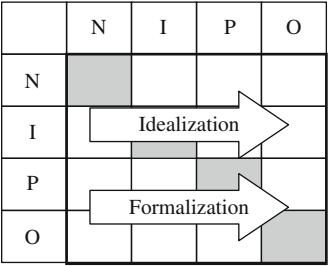


Fig. 5 Abandonment and informalization

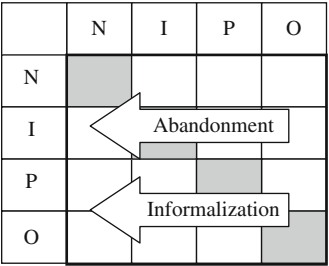
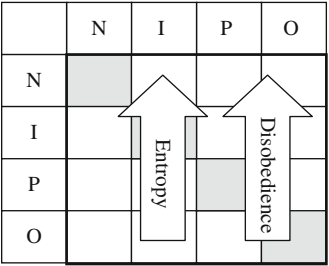


Fig. 6 Entropy and disobedience



provider (G) explained that its change and project management practices were shaped by experiences and not written and defined explicitly. According to the interviews, these practices were, however, uniform across the whole organization. The following is a typical statement representing emergence of a practice:

To my knowledge, we have no ready-made method taken outside the organization. Instead, software development methods and practices and how things are done are grown here during the years. (Project Manager, Case G)¹

This emergent nature of methods is further discussed by the same project manager:

We have created such a ... it is mostly unwritten rule ... it works in our environment ... everyone knows it, but we have not put it in a written form, we have discussed it, but the rule determines how we act and what we do for example when we change software and it states how we do the specification and changes and commenting, testing and so on. (Project Manager, Case G)

Adoption (Fig. 3) represents a typical idea of how systems development innovations diffuse in organizations [11]. A practice is planned and taken into use through rational management actions. The developers are required to enact a defined and, usually, documented practice. An obvious factor promoting this type of change is the loyalty and obedience of the employees. In organization D all projects used a specific set of development practices, Scrum [17], because it was company policy. One developer said he was more likely to follow defined practices if the management was strict. In F the enterprise defined the methods, but they were in a constant process of adoption:

Our company offers methods, but we had no experience on those methods and how specifications should be made, what will be the result, and what should be the quality level of the documentation. Therefore even when we made the specifications the whole last year, eventually we were in a hurry and we did not succeed ... we can now say that the specifications were not at a level detailed enough. (Department Manager, Case F)

In *Idealization* (Fig. 4), an organization makes a decision to introduce, or increase the scope of a practice. Typically, the decision is made by a manager or other stakeholder with the power to suggest changes. The practice may be part of an ISD method, or it may be enacted by an individual or by a project and the management considers it worth of broader use. The practice is then explicitly defined, possibly documented, but not yet necessarily adopted, which requires the process of Adoption.

In the data, typical reasons for idealization included expectations of better quality, complexity management, and reduction of risks through better practices. An example is organization A, which was going to define a testing practice by the time of data collection:

Yes, there is one area in which we have thought to introduce methods, which is testing and approval. Here we plan to formalize something. A third person can go in to do a quality check and to test. Sometimes the quality produced by developers may vary a bit. (Senior Developer, Case A) (Note that despite the interviewee mentioning the term formalization here, the company was going to initiate and idealize a new practice for testing, not to formalize an existing one as is the case below.)

¹Most of the quotations are translated from Finnish or Norwegian.

Formalization (Fig. 4) means that an organization decides to formally accept a widely enacted practice. The practice may have spread through the process of emergence, and formalization is acceptance of the fact. For example, in Case I, requirement specification practices had emerged over time without explicit practice definitions. Both customer pressures and their own organizational development required them to start formalization of the already existing requirement specification practices:

They wish that we define the ways how they can participate to the process. For example, the developers at the customer are doing many tasks for us. Probably half of our designs come directly from the customer – they do it that way – and then would very much better understand how they should do it, because they are not engineers. (Upper Manager, Case I)

Abandonment (Fig. 5) means that an organization decides to accept the fact that a predefined practice is not enacted, in a sense the opposite of Idealization. The reasons for this may include that the practice does not work or that it does not give enough benefits in practice. From the data we could find several examples of opinions that the practices do not work or that there are not enough perceived benefits of practices. This can lead to a situation where a practice is silently abandoned or a decision is made to abandon it. The following excerpt from an interview of a Case H developer exemplifies this:

Q: Do you have guidelines or common practices related to systems development in your company?

A: We have software development process in our company. This process is ISO certified. We have number of documents which we follow. The project should have a special format, documents, discretion of the process what stages the project should have.

Q: How strictly do you follow these documents?

A: We are trying to follow but how strict as I said depends on the customer because always following documents requires additional time for management and in some cases we do not have this time. So it is agreed with the customer what project documentation will be used for the particular project.

Informalization (Fig. 5) happens when the organization decides to stop using, or reduce the scope of, a defined practice that has previously been widely enacted and formalized. This may happen because the practice in question is not seen as important anymore, or because an organizational change removes the sponsors of the practice. An example is case C, a former IBM subsidiary that had largely continued their practices, in spite of lack of management attention:

There is no pressure to follow the guidelines any more. It was different in IBM. (Developer, Case C)

The company later went through a formalization process which reintroduced the practices as official policy:

We have written a quality manual that is based on the practices carried over from the IBM time. (Project Manager, Case C)

Entropy (Fig. 6) means that a practice that was widely enacted before is no longer followed despite its intended normative scope. This may happen, e.g., because the practices do not fit to new kind of tasks or because the individuals do not feel confident with the practices. In Case I the quality that the practices produced was in some projects considered as too high and expensive and therefore some decided not to use the defined practices.

I would say that we expect too high quality, because testing and developing currently take quite a time. [...] I think our quality is probably too good. . .for our customers' needs, we are just . . .how to say it. . .we take too much pride perhaps in that it works. (Manager, Case I)

Disobedience (Fig. 6) is a special form of Entropy, where an actor or a project does not enact a practice, when a decision has been made to adopt it. Possible reasons, in addition to those mentioned above, include opposition to the management and lack of training of new employees. In organization B at least one developer was not aware of well-defined and documented practices that he was supposed to follow. A slight scent of disobedience can be observed from the following comment from Case F as well:

This has been a rather crude experience as well. [...] We did the specification with [the defined method], but I have not familiarized myself more to that, but I guess it is some kind of process or a model of how the specification is executed and what kind of documents are produced in the specification. People tried to learn that and went to some kind of courses. Somehow I got the picture that it did not guarantee anything and it would have forced us to do this and that kind of tasks. (Manager, Case F)

6 Discussion

In the previous section, we summarized eight stereotypical change paths of systems development practices identified in our target organizations. We argue that altogether they represent an integrated view on studying how systems and software development practices are shaped in organizations. The identified change paths complement previous research, such as:

- previous theories and models that have explained change in ISD process innovations, such as the diffusion of innovations theory (DoI) [11, 16], or the models for software process improvement [5] and capability maturity [13],
- the “snapshot” frameworks of ISD practices in organizations, such as the method-in-use framework suggested by Fitzgerald et al. [3], and
- causal factor analyses of issues having impact on adoption of systems development methods and practices (e.g., [4, 6]).

The DoI theory explains ISD process innovations through two concepts: initiation and implementation [11]. These correspond roughly to the idealization and

adoption concepts of our framework, respectively. Mustonen-Ollila and Lyytinen [11] find that DoI cannot explain all ISD process innovations. In addition to the initiation (idealization) and implementation (adoption), we identified six additional change paths. These include change paths which describe how some innovations come to their end, are abandoned, and how new ones may emerge from individual initiatives even without much planning. In this sense, we argue that the eight identified change paths involve more explanatory power forming a potential basis for more detailed theoretical elaborations about how ISD practices change in organizations.

Software process improvement and the capability maturity model [5, 13] seem to suggest that change in practices proceeds toward an improved software process or greater maturity. However, our data and four of the change paths illustrate that ISD practices may come to end, as well. We argue that our model has a greater explanatory power to illustrate that than the normative models focusing only on improvement and innovation. Not all change is for the better, for various reasons.

The method-in-action framework by Fitzgerald et al. [3] has been used to describe how ISD practices emerge in context (e.g., a development project, cf. [10]). However, it is a high-level view that merely acknowledges that actual practice is shaped by contextual factors. The NIPO grid and the change paths supplement the method-in-action framework by providing concepts that can be used to break down complex change paths, and describe and analyze them in detail. In particular, it separates analysis of decision processes (the horizontal paths) from the enacted practice (the vertical paths).

Compared to previous empirical studies of causal factors which affect on method and practice utilization in development organizations, e.g., [4, 6], we argue that the suggested framework provides a more in-depth and nuanced view of the situation. At different times, different issues having potential impact on status quo of the contemporary practice may gain momentum and change a practice in use. Such phenomena are not easy to find through factor-oriented analysis. Based on our results, we expect that it may appear useful to understand in-depth patterns of how practices change in development organizations in light of the NIPO grid. For example, one organization may recognize that testing practices have first emerged and defined afterwards, whereas another organization may have first idealized their testing practices, after which they have been adopted to a project or the whole organization. In such a situation, the organizations would have followed different change paths to a similar state of art: standardized and enacted testing practices. However, factors which may affect a successful idealization – adoption may be different from those related to emergence – formalization, and too simple factor-based cause – effect explanations may not be able to explain the end results, hence the argumentation for our model giving a more nuanced view.

Our study is not without limitations. First, the elaboration of the framework of change paths is mainly conducted through identifying examples from the data and we have compared our results to the literature retrospectively. More rigorous literature categorization is needed (although not feasible within the conference paper space limits). We also wish to take another round of data collection now focusing on

the changes in practices from beforehand to confirm our results with better “selective coding,” and with more concise case studies about particular practices in particular organizations. Although we cannot guarantee the theoretical saturation of our data collection yet, we identified a theoretically representative set of change path types on the NIPO grid from the data. That is, all directions toward which a practice can change on the NIPO grid are covered with examples. There are obviously many possibilities for more fine-grained analysis and richer variation of concepts which might be picked to describe particular change path types in more detail that do not come up in our observations. But we still believe that the study shows enough saturation for the identification of the eight change path types at a general level.

Our ultimate aim to build up theories about how changes happen on particular types of systems and software development practices is still to be reached. However, we consider our framework to provide a promising basis for theorizing about how ISD practices change. Longitudinal follow-ups of change in particular practices, explaining reasons and paths for change in light of our framework, remain in our future research agenda.

With more focused data from the field, we also plan to examine possibilities to abstract from the field of ISD practices toward theorizing about change in organizational practices in general. However, this requires more work with the literature about organizational change in general and an adjustment of our NIPO framework – taken that not all organizations are project organizations.

7 Conclusions

In this chapter, we studied the issue of how ISD practices in systems and software development organizations are shaped. As an answer to the question, we identified eight stereotypical change paths, through which a particular practice may evolve or be developed toward another placement on the NIPO grid: emergence, adoption, idealization, formalization, abandonment, informalization, entropy, and disobedience. Change in an ISD practice is conceptualized through two dimensions. That is, an ISD practice may change with regard to its definition (scope and content of planned practice) and/or with regard to its enactment in practice. We claim that the resulting framework complements the current models and theories applied to studies of ISD practice change and innovation.

References

1. Brown, J. S., and P. Duguid (1991) Organizational Learning and Communities-of-Practice. Toward a Unified View of Working, Learning and Innovation. *Organization Science* 2 (1): 40–57.
2. Collins COBUILD English Dictionary (1989).
3. Fitzgerald, B., N. L. Russo, and E. Stolterman (2002) *Information Systems Development: Methods in Action*. McGraw-Hill Education, London.

4. Hardgrave, B. C., F. D. Davis, and C. K. Riemenschneider (2003) Investigating Determinants of Software Developers' Intentions to Follow Methodologies. *Journal of Management Information Systems* 20(1): 123–151.
5. Humphrey, W. S., T. R. Snyder, and R. R. Willis (1991) Software Process Improvement at Hughes Aircraft. *IEEE Software* 8(4): 11–23.
6. Khalifa, M., and J. M. Verner (2000) Drivers for Software Development Method Usage. *IEEE Transactions on Engineering Management* 47(3): 360–369.
7. Kogut, B., and U. Zander (1992) Knowledge of the Firm, Combinative Capabilities and the Replication of Technology. *Organization Science* 3(3): 383–397.
8. Kostova, T., and K. Roth (2002) Adoption of an Organizational Practice by Subsidiaries of Multinational Corporations: Institutional and Relational Effects. *Academy of Management Journal* 45(1): 215–233.
9. Larsen EÅ, Päivärinta T, and Smolander K (2008) The Nipo Grid – A Construct for Studying Systems Development Practices in Organizations. In 16th European Conference on Information Systems (Golden W, Acton T, Conboy K, van der Heijden H, Tuunainen VK eds.), 398–409, Galway, Ireland. (ISBN 978-0-9553159-2-3)
10. Madsen, S., K. Kautz, and R. Vidgen (2006) A Framework for Understanding How a Unique and Local IS Development Method Emerges in Practice. *European Journal of Information Systems* 15(2): 225–238.
11. Mustonen-Ollila, E., and K. Lyytinen (2003) Why Organizations Adopt Information System Process Innovations: A Longitudinal Study Using Diffusion of Innovations Theory. *Information Systems Journal* 13: 275–297.
12. Nelson R., and S. Winter (1982) *An Evolutionary Theory of Economic Change*. Belknap Press, Cambridge, MA.
13. Paulk, M. C., B. Curtis, M. B. Chrissis, and C. V. Weber (1995) *Capability Maturity Mode: guidelines for improving the software process*, Addison-Wesley, Reading, MA.
14. Päivärinta, T., M. K. Sein, and T. Peltola (2010) From ideals towards practice: paradigmatic mismatches and drifts in method deployment. *Information Systems Journal* 20(5): 481–516.
15. Pentland, B. T., and M. S. Feldman (2005) Organizational Routines as a Unit of Analysis. *Industrial and Corporate Change* 14(5): 793–814.
16. Rogers E. M. (1995) *Diffusion of Innovations*, 4th edn. The Free Press, New York.
17. Schwaber K., and M. Beedle (2001) *Agile Software Development with Scrum*. Prentice Hall, Upper Saddle River, NJ.
18. Strauss, A. L., and J. M. Corbin (1990) *Basics of Qualitative Research: Grounded Theory Procedures and Techniques*. Sage, London.
19. Szulanski G. (1996) Exploring Internal Stickiness: Impediments to the Transfer of Best Practice within the Firm. *Strategic Management Journal* 17(Winter Special Issue): 27–43.

Designing a Study for Evaluating User Feedback on Predesign Models

Jürgen Vöhringer, Peter Bellström, Doris Gälle, and Christian Kop

Abstract Predesign schemata are an attempt to overcome the communication gap between end users and computer scientists in software development. We argue that by presenting facilitated models to the users, schema interpretation is made easier. The predesign principle is also helpful in areas such as software documentation. Our uses of predesign models share the basic assumption that these models are easier to understand for end users. In this chapter we present a design for an experimental study that aims to evaluate these assumptions.

Keywords Modeling · User feedback · Experimental study

1 Introduction

Communication between system developers and domain experts has always been prone to mistakes and misunderstandings in system development. Modeling languages like UML (Unified Modeling Language) [7] are used for abstracting the real world, though domain experts may still have problems giving adequate feedback. The reason is that the interpretation of non-predesign schemata can be difficult without deep background knowledge about modeling, which most domain experts lack as they have no information on systems education. Predesign schemata are an attempt to overcome this gap between end users and computer scientists in software development. By presenting facilitated models to the users, schema interpretation is made easier. In [9] we presented our own predesign model. The predesign principle is also helpful in other areas such as documentation for end users. However, our uses of predesign models share the basic assumption that these models are easier to

J. Vöhringer (✉)

Institute for Applied Informatics, Research Group Application Engineering, University of Klagenfurt, Klagenfurt, Austria

e-mail: juergen@ifit.uni-klu.ac.at

understand for end users. We present a design for an experimental study that aims to evaluate these assumptions.

The chapter is structured as follows: In Section 2 we motivate the predesign approach and explain the predesign principles. We show that the predesign approach is language-independent and can be applied to modeling languages like UML. In Section 3 we give examples for typical application areas where predesign models can be utilized successfully. In Section 4 we describe the design of an experimental study that aims to evaluate the easy understandability of predesign models, which is one of our main motivations for working with predesign models. Finally, in Section 5 the chapter closes with an outlook to future work.

2 What Is Predesign Modeling?

A predesign model is a conceptual model, which is usually generated in the early phase of conceptual design, right after requirements analysis. The name predesign was chosen since this kind of model can be seen as a first sketch for the conceptual design model. It is located on a level closer to natural language compared with more traditional modeling languages like UML, which are often hard to understand for end users. Predesign models on the other hand try to facilitate the communication with the end users. As a consequence the predesign principle is based on the following ideas:

- Only those modeling concepts are selected which are common to the different kinds of conceptual modeling languages.
- Lean set of modeling notions.
- Graphical and glossary representation.

A conceptual predesign modeling language is designed based on the idea to take those modeling constructs which are shared by many conceptual modeling languages. Even though it is possible to use modeling languages developed and designed for predesign, e.g., the Klagenfurt Conceptual Predesign Model (KCPM) [9] or the Enterprise Model (EM) [6], any modeling language can be abstracted to follow the predesign principles discussed in [2].

For static modeling, this means the following: modeling languages like UML, Entity Relationship (ER) Diagrams, and Object Role Modeling (ORM) essentially all show terms (in KCPM called thing types) which are related to each other (in KCPM through so-called connection types). These most essential static concepts are thus used in static predesign modeling, while more sophisticated modeling techniques are not required. Table 1 shows a mapping of these essential modeling elements.

For specifying behavior of a system through dynamic models, modeling languages often work with notions that can express that somebody/something (i.e., a concept from the respective static model) executes something. The execution (in

Table 1 Essential concepts for static models

Modeling language	Thing type	Connection type
Class diagram	Class, attribute	Association
ER	Entity, attribute	Relationship
ORM	Entity/label	Role

[2] called process type) is done under certain preconditions. After the execution certain postconditions hold true. Different modeling languages for modeling behavior (i.e., state charts, activity diagrams, business process models, etc.) have different strategies and modeling constructs to model these things. Depending on the concrete modeling language, some of the above principles are modeled in more detail than others and have different features. Like in the static realm, dynamic conceptual predesign languages focus only on the most essential principles and try to treat them equivalently. Table 2 shows a mapping of these essential dynamic modeling elements.

Hence a conceptual predesign modeling language only uses modeling notions that are absolutely necessary, i.e., mainly the ones shown in Tables 1 and 2. For instance, in static modeling, both standard UML class diagrams and ER diagrams distinguish between class (entity types) and attributes. Predesign models do not make this distinction however, since this is an implementation-dependent decision that is not important in an early phase of software engineering. When mapping predesign models to standard design models such a distinction can often be made automatically. A simplified version of UML could be used for this purpose, especially given UML’s position as a de facto standard for object-oriented modeling [13] and its widespread use during conducting requirements engineering [8]. We argue that applying the predesign principles on UML models offsets the complexity and heavy focus on implementation issues which can be seen as a major weakness of standard UML models [14].

However, predesign models should not only be facilitated according to the used modeling elements, but their contents should also be design-independent, describing only domain without forestalling implementation details. This also includes concept designations which should be as self-explanatory and easily understandable as possible. Finally, the representation technique depends on the capabilities of the end user. Therefore a predesign language should allow different representation techniques for the same model like graphical schemata or glossaries.

Table 2 Essential concepts for dynamic models

Modeling language	Process type	Pre-/postcondition
Activity diagram	Activity	Edges , object flows, conditions
State chart	Transition	State
Petri net	Transition	Place

3 Application Examples for Predesign Models

As shown in Section 2, one reason for working with predesign models is their lower complexity compared to regular conceptual schemata. On the other hand predesign models are more formalized and therefore less ambiguous than natural language, which also enhances understandability. This makes predesign models ideal tools to use whenever communication with end users or domain expert feedback is required. In the following we describe several areas where predesign models are currently applied successfully: *requirements elicitation*, *schema integration*, and *software documentation*.

During requirements elicitation, a glossary representation is useful. Such glossaries have the advantage of being like check lists. In the rows, the domain notions are collected. In the columns additional information for each domain notion is gathered. If there is an empty cell in one of the rows and columns, then both end users and designers get easily aware that something is missing. For certain types of end users, e.g., business economists, glossaries are also easier to understand than graphical models and therefore facilitate discussion of the requirements with these stakeholders [9].

Schema integration is another important field of application for predesign models. Combining requirements from different sources in a software development project leads to an integration problem, as commonalities and differences must be identified and resolved. As discussed in [16] integration on the predesign level has both advantages and limitations that distinctly separate it from other integration approaches that were developed for more formal models. We argued that the advantages – in particular the opportunity to get better user feedback on the integration due to simpler schemata that are easier to grasp intuitively – outweigh the limitations. In [1] an integration process for predesign schemata was proposed. In several phases of the process end users have the possibility to intervene and give their own feedback. Expert user feedback is explicitly encouraged for decision guidance whether two compared concepts match or for resolving conflicts during resolution. The integration process is designed so that users are presented with automatically calculated matching and integration proposals, any possible warning (e.g., concerning homonym and synonym conflicts), and any additional concept metadata which was not used for calculating the automated proposal (e.g., examples of instances). The output of the comparison process is a list of matching decisions which can then be processed by automated integrated steps for all concepts and the relationships connected to them, resulting in the integrated schema, which can and should be evaluated by domain experts again. Thus, the domain experts have a direct influence on matching decisions and the integration result.

Predesign models are also expedient for documentation of already implemented software. Ideally the predesign models developed in the early phases of a software development process are reused as part of the documentation after the project has been finished. The need for understandable documentation increases if the software and its workflows are not self-explanatory. Web services are a typical example for non-self-explanatory software. They offer a certain functionality that can be

integrated in existing software. A Web service is meant to be invoked by other software and so the documentation is formatted in machine-readable XML code (WSDL [17], OWL-S [11]). However the selection of the right service is still a job for humans, especially if the services are publicly available. An easily understandable Web services documentation helps domain experts choosing the right service for their requirements. They need information about the structure and processes of a service. This information could be given in natural language, which is less formal and ambiguous and the information is sometimes scattered among the text. As discussed in [3], important Web service information can be shown in glossaries (i.e., predesign models). Some advantages are that sophisticated information can be hidden from the domain experts (e.g., information about the invocation) and information belonging together (e.g., operation name, parameters, and data type) can be represented in a concentrated easy understandable way. Dynamic models are also needed for documenting Web service processes. The aim is to give the readers a good overview of the functionality that can be understood without further investigations in the IT respectively Web service domain. Predesign models can help to achieve this aim. As stated in [3], there are other possible readers of Web service documentation beside domain experts. However, predesign models can also be used by more sophisticated readers like software engineers or Web service architects as they show a good overview about Web service structure and processes.

4 An Experimental Study on Predesign Model Understandability

In 2000, [15] surveyed how predesign models are approved in small- and medium-sized Carinthian enterprises. The survey did not try to objectively measure the understandability of the models however. Furthermore, it mostly dealt with the KCPM glossary view, which is very useful for certain customers and problems (i.e., during requirements elicitation and documentation), but in other situations a graphical representation of the same predesign model might be more efficient. In this section we describe the design of an experimental study that we are currently performing and that addresses some of the shortcomings of the previous surveys. Thus the study aims to formally evaluate some of the assumptions that are a basis for our ongoing predesign research. Similar studies testing understandability and modifiability of UML models can be found in [4, 5]. Also, [13] compared the understandability of UML class diagrams and use cases.

4.1 Hypothesis

The main hypothesis to be tested in this study is that predesign models facilitate interpretation for domain experts who have no extensive modeling knowledge. A secondary hypothesis, inspired by the survey results of [15], is that dynamic

models are generally more difficult to interpret for domain experts than static models. Specific aims of the experimental study are:

1. Determining if there is a significant difference in subjective experiences about the understandability of predesign and non-predesign models.
2. Determining whether an objective difference in correctness between the interpretation of predesign and non-predesign model can be measured.
3. Determining whether the subjective and objective study results concerning predesign model understandability match.
4. Determining if a similar discrepancy in subjective and objective understandability can be measured for static and dynamic models.

Aims 1–3 correlate to hypothesis 1, while aim 4 correlates to the hypothesis 2.

4.2 Experimental Design

The study is an experimental research, where we monitor one experiment group and one control group. Participants of the experiment group are shown a survey containing two models following the predesign principles – a static model and a dynamic one. Questions both about subjective experiences with the models and objectively assessable interpretation of the models are posed. Likewise, the control group is presented with non-predesign static and dynamic design models that are closer to implementation and use extended modeling concepts. Thus, the study follows a typical posttest-only control group design:

- R x T1 O
- R x T2 O

R means that participants are randomly assigned to groups. x means that no pretesting is performed. T1 and T2 are the two treatments for group 1 (experimental group) and group 2 (control group), respectively. The treatments are the different models that are shown to the participants. O means that participants are posttested on the dependent variable, i.e., the model understanding. The independent variables of the study are the category (with the values “predesign” and “non-predesign”) and the type (with the values “static” and “dynamic”) of the models. The dependent variables are the subject’s objective performance in interpreting the requirements depicted by the models and their subjective answers (both quantitative and qualitative) (Table 3).

4.3 Subjects and Domain

A pilot study is currently taking place at the University of Klagenfurt. Both the experiment and the control group consist of participants who have no deep prior

Table 3 Relevance of dependent variables for study aims

	Aim 1	Aim 2	Aim 3	Aim 4
Objective performance		X	X	X
Subjective answers (quantitative)	X		X	X
Subjective answers (qualitative)	X		X	X

knowledge about modeling or software development: the subjects are business economy students who are attending an introductory modeling course. They have been taught the basics of static and dynamic UML models over the course of 2 months and were then randomly assigned to the experiment or control group. We do not expect the participants’ previous UML knowledge to bias the study, since the study does not aim to evaluate the benefits of UML but the benefits of abstracting implementation and design details in predesign models. After the pilot study we will however repeat the study with users that do not have any previous modeling knowledge.

Figures 1 and 2 show the static predesign and non-predesign models used for the current study. The dynamic models depict the processing of an order that arrives in a company. The static models show the structure of the same enterprise. This domain was chosen deliberately since it should be familiar for the business economists, who are the main participants in the study. Subtleties about this specific domain can only be derived from the models themselves however, which makes understanding the models crucial for answering the follow-up questions. The non-predesign models were developed from practice examples of UML models. The predesign models

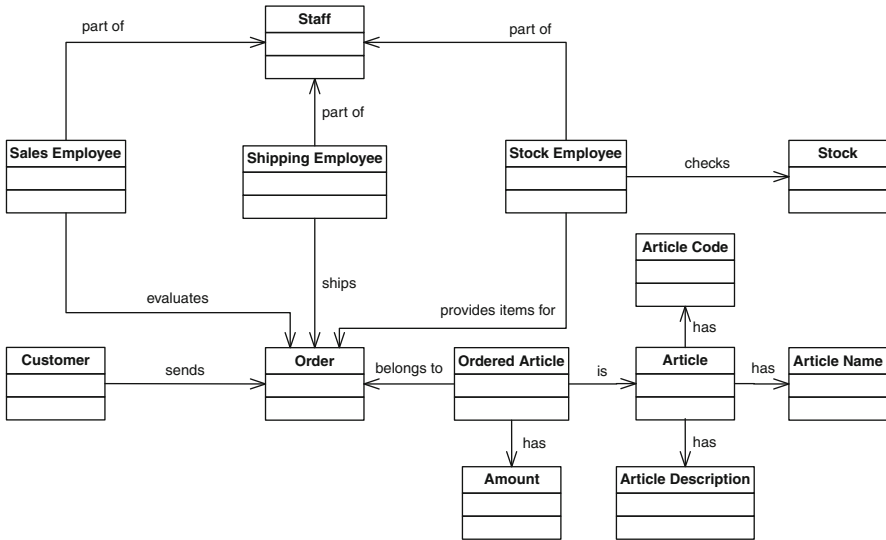


Fig. 1 Predesign version of the employed class diagram

students to this course, they were given access to the study. The predesign and non-predesign models are both available in German language in order to prevent the language barrier from negatively influencing the interpretation of the models – past experiences have shown us that many students in early stages of their studies have difficulties with understanding English texts. The students have 2 months time for participating in the study. The study takes about 1 h of time to complete and students have only one try – after submitting it the survey becomes inaccessible for them.

As modeling languages for the non-predesign schemata we chose UML 2 class diagrams and activity diagrams, because both are used frequently in practice. Also, since at least the pilot study is performed on students who have some basic knowledge about class and activity diagrams (but not about predesign modeling or extended UML 2 modeling concepts), it might motivate them to engage more earnestly in the study. For the same reason we refrain from using a special predesign modeling language like KCPM, but instead we employ facilitated UML class and activity diagrams (see Section 2), which adhere to our predesign directives.

4.5 Question Design

The questions of the study are a mix of quantitative, qualitative, and follow-up questions. In [12], p. 111, it is argued that a mix of both quantitative and qualitative questions can be useful in a survey under certain circumstances, i.e., when open questions are used for explaining and motivating closed ones.

We use quantitative questions to ask the participants about their subjective attitude regarding the models. Do they seem complex to them? Do the students feel such models are suitable alternatives or enhancements for natural language domain descriptions? These kinds of questions are presented as multiple choice. Since the test is performed via an eLearning platform we can and have to ensure that only one answer is allowed and that no answers are preselected when starting the test. Moreover, as mentioned in [12], p. 161, when participants have to choose between contrasting alternatives in multiple choice questions, people tend to choose the “safe” middle position. In our survey, we therefore provided mostly even numbers of answering alternatives, offering no middle position. An alternative might be mixing the sequence of the answers, so that the middle answer is not always the neutral one. This also forces participants to read questions thoroughly.

Qualitative questions are harder to evaluate than quantitative ones, because they rely on free text answers by the participants. These kinds of questions have the advantage however that participants are not restricted on few expected answers and thus they are not indirectly manipulated by the study conductors. Some participants tend to think out of the box and give answers the study designers would not have thought of themselves. We use free text questions for involving the participants more directly in the study and for providing us with additional information about the quantitative answers: for instance, participants are asked to explain and motivate their multiple choice answers and give improvement proposals for models, and we also ask them whether they used external help like searching the Internet or

support from friends when participating in the online study. We do not reject the use of external help, though we also do not explicitly endorse it. On the contrary we believe that many students will not put up with the extra effort of online research or getting help by friends, even if they have the chance. Our experience from previous requirements engineering projects showed us that many people simply do not have the time or patience to do this extra work, even if it would help them to understand models and facilitate communication with the system designers. Thus, we believe that while it is realistic to provide study participants the principle possibility to get external information if they want to, we do not expect this opportunity to be seized frequently by the participants.

Follow-up questions finally are a special form of qualitative questions that demand free text answers about the contents of the models. These answers can be classified as correct or not correct, in contrast to qualitative questions that are open-ended and have no correct answer. Answers to follow-up questions are therefore used for tracking whether the participants have really understood the model. Table 4 lists some questions from the experimental study. An identical set of questions is used for both the experiment and the control group. The predesign and non-predesign models share the same contents but represent it in a different way, so the same follow-up questions are used for them. The static and dynamic models for each group also have identical qualitative and quantitative questions, but the follow-up questions differ naturally.

In summary, the quantitative data provide measurements for the subjective attitude of the participants and the data gathered from the follow-up questions provide measures for the objective measures of performance. Additionally to standard quantitative, qualitative, and follow-up questions we also pose some demographic questions in the beginning of the survey, i.e., age group or gender of the participants. These demographic questions are mostly multiple choice, though there is also one allowing free text answers. These demographic data are not really relevant for the study itself (they are not counted among the independent variables), but in combination with the results they might hint on additional variables that are worth researching in future studies.

Table 4 Exemplary questions

Question	Type	Models	Goal
What was your first impression of the model?	Multiple choice	Static and dynamic	Subjective impression of model understandability
Who is responsible for checking the status of the stock?	Follow up	Static	Evaluation of actual model understanding
How would you improve the above model? Which parts should be facilitated?	Free text	Static and dynamic	Get constructive criticism on models

4.6 Evaluation Methods

Several strategies exist for evaluating quantitative research data. Statistical significance is the standard approach, though it requires a large number of subjects to get statistically significant results. The sample size of our pilot study is about 30 participants for each group, which is rather small for a reliable statistical analysis; future studies will allow us performing according tests however. In our pilot study we plan to use more accessible methods like standard deviation and chi-squared tests for determining the effect of using predesign models on user interpretation and feedback. These techniques will be applied on all multiple choice questions of our study, besides demographic questions which only provide additional context to the survey results.

For the evaluation of text answers in qualitative questions we intend using an approach from social sciences. Qualitative content analysis focuses on assigning fragments from open-ended answers to categories [10]. The determination of these categories is quite difficult and there are two main approaches for obtaining them. In the deductive approach the categories are defined on the base of assumptions and theories before the text is analyzed. In the inductive category development, strategy categories are created on the base of the texts. We believe the latter approach to be better suited for our application area. We aim to get an overview for possible problems and advantages with the use of predesign models. If we define the categories in advance, maybe there are some perspectives that get ignored.

Answers to follow-up questions are given in free text but they only need to be categorized into right or wrong answers. These answers can be utilized for supporting or refuting the quantitative answers regarding model understanding.

4.7 Future Work

To get optimal results and minimize the influences from unexpected external factors we plan to perform several iterations of the study under slightly different circumstances. It is planned to repeat a variation of the study in summer for participants who have no modeling and information systems background at all. Also, we plan to perform the study in Swedish language with information systems freshmen at Karlstad University in Sweden.

5 Summary and Conclusion

A pilot study following the design presented in Section 4 is currently performed at the University of Klagenfurt. While the quantitative results on the multiple choice questions will give us information about the participants' subjective opinions regarding the understandability of the various models, evaluation of the follow-up questions will give us more objective indication whether the participants have understood the respective models.

The results of the study will be discussed in future publications. In order to get a bigger pool of participants and eliminate unintended external factors, the study will be repeated with freshman students at Karlstad University in Sweden. By that we hope to increase the reliability of our results. We also plan to perform a version of the study with participants that have no prior modeling knowledge. In subsequent studies the used models will be adapted to the new situation. Further research steps will depend on the results of these studies. Our research regarding the application of predesign models in different contexts will also continue.

References

1. Bellström P., and Vöhringer J. (2009) Towards the Automation of Modeling Language Independent Schema Integration, In *International Conference on Information, Process, and Knowledge Management*, eKnow 2009, pp. 110–115.
2. Bellström, P., Vöhringer, J., and Kop, C. (2008) Guidelines for Modeling Language Independent Dynamic Schema Integration, In: Pahl, C. (ed), *Proceedings of the IASTED International Conference on Software Engineering*, ACTA Press, pp. 112–117.
3. Gälle, D., Kop, C., and Mayr, H. C. (2008) A Uniform Web Service Description Representation for Different Readers. In *Proceedings of the Second International Conference on Digital Society*, ICDS, IEEE Computer Society, pp. 123–128.
4. Genero M., Piattini M., and Calero C. (2002) Empirical Validation of Class Diagram Metrics, In *Proceedings of the 2002 International Symposium on Empirical Software Engineering*.
5. Genero M., Piattini M., and Manso E. (2004) Finding “Early” Indicators of UML Class Diagrams Understandability and Modifiability, In *Proceedings of the 2004 International Symposium on Empirical Software Engineering*.
6. Gustas R., and Gustiené P. (2004) Towards the Enterprise Engineering Approach for Information System Modelling Across Organisational and Technical Boundaries, In *Enterprise Information Systems V*, pp. 204–215.
7. Hitz, M. et al. (2005) *UML@Work*. dpunkt.verlag, Heidelberg, ISBN 3-89864-261-5.
8. Kaindl, H. (2005) Is object-oriented requirements engineering of interest? *Requirements Engineering* 10, 81–84.
9. Kop C., and Mayr H.C. (1998) Conceptual Predesign – Bridging the Gap Between Requirements and Conceptual Design, In *Proceedings of the ICRE’98*, pp. 90–100.
10. Mayring P. (2000) Qualitative Content Analysis. *Forum Qualitative Social Research* 1(2), Art. 20.
11. OWL-S (2004) *Semantic Markup for Web Services*, W3C Member Submission, 22 November 2004, <http://www.w3.org/Submission/OWL-S/>
12. Schuman, H., and Presser, S. (1996) *Questions & Answers in Attitude Surveys Experiments on Question Form, Wording, and Context*, Sage Publications, Inc.
13. Siau K., and Lee L. (2004) Are use case and class diagrams complementary in requirements analysis? An experimental study on use case and class diagrams in UML. *Requirements Engineering* 9, 229–237.
14. Siau, K.L. (2004) Theoretical and Practical Complexity of UML, In *Americas Conference on Information Systems*.
15. Stark M. (2000) *Geschäftsprozessmodellierung im konzeptuellen Vorentwurf*, Diploma Thesis, Institute for Applied Informatics, Research Group Application Engineering, University of Klagenfurt, Austria.
16. Vöhringer, J., and Mayr, H.C. (2006) Integration of Schemas on the Pre-Design Level Using the KCPM-Approach. In Nilsson, A.G. et al. (eds), *Advances in Information*

- Systems Development: Bridging the Gap between Academia and Industry*, Springer, pp. 623–634.
17. Web Services Description Language (WSDL) Version 2.0 Part 1 (2007) *Core Language: W3C Recommendation*, 26 June 2007; <http://www.w3.org/TR/wsdl20/>.
 18. Williams, B. (2005) *Moodle for Teachers, Trainers and Administrators*, V.1.4.3, Manual, <http://download.moodle.org/docs/en/using-your-moodle.pdf>.

Study on the Method of the Technology Forecasting Based on Conjoint Analysis

Jing-yi Miao, Cheng-yu Liu, and Zhen-hua Sun

Abstract We discuss an application of conjoint analysis in technology forecasting, summarize basic operation steps of conjoint analysis, and give a stimulant example of technology forecasting. In this example, we consider five factors that will affect the emergence of a new technology. These factors have investing demand in a new technology, potential market value of a new technology, realizable difficulty of a new technology, supporting degree of relative technology to a new technology, and the competitive power of a new technology with original technology. Technology development has a discontinuity. With discontinuity, we cannot forecast the future of technology development, based on the current trend of technology development. As using quantitative methods to make forecasting, we assumed that current trends of technology development hold a fixed law, so those quantitative methods cannot forecast the discontinuity of technology development. Some subjective forecasting methods have huge improvement in technological discontinuity forecasting. The improvement is that forecaster's subjective judgments and capability are embodied in forecasting. But this method has two inherent defects: one is the lack of design ability, which makes this method susceptible to the influence of organizer and forecasting, and the other is that while facing numerous forecasters, the forecasting data are often difficult to explain and analyze; we also have difficulty in making a synthesized judgment. A subjective and synthesized judgment of technology development is similar to economical utility, thus we could apply the measure of colony's utility to improve the appropriateness and reliability of subjective forecasting method. Using conjoint analysis, we can judge the colony's utility accurately, because the datum that we use in analysis comes from the subjective judgments of forecasters to various fields of technical development, but the influence of the random error can be dispelled by using some theory model about data processing. Therefore, conjoint analysis method is one useful tool for technology forecasting. The effectiveness of this method has been testified by simulation experiment.

J.-y. Miao (✉)

School of Management Science and Engineering, Shanxi University of Finance and Economics, Shanxi, China

e-mail: jingyimiao@126.com

Keywords Conjoint analysis · Colony's utility · Simulation · Technology forecasting

1 Introduction

Rapid scientific progress has vastly increased the degree of complexity of inter-related technologies. A systematic and comprehensive approach is required for forecasting such technological changes. We have to evolve tools which can cope with the complexity of interrelationship in technology, variability of events, and lack of information about the future. Technological forecasting techniques provide some such tools.

Under the guidance of scientific theory and method, technological forecasting starts from the history and current situation of the technology development; conducts qualitative and quantitative analysis to study the principle of technology development, and then forecasting its trend. For obtaining those study results, the study should make a deep qualitative analysis and strict quantitative calculation to the process of the technical development based on statistics and survey materials.

Technological forecasting is a main content in an enterprise's technological strategic management frame; it is an indispensable important method and tool in technological strategy. The task of technological forecasting does not involve discovery and improvement of technology, however, forecasting could help people understand the influence of future technology invention on society, economy, and technology development. The aim of technological forecasting is to plan for the future research and development program through perceptive study and analysis of current and future conditions and environments. It is usually concerned with forecasting the occurrence and effect of events which imply fundamental change in technological and social environment.

Xu [1] built a categorized tree of technological forecasting methods according to Worlton's category [2]; the category divided technological forecasting methods into two basic categories, one is exploratory methods and the other is normative techniques.

Bahl and Arora [3] summarize traditional methods of technological forecasting. Exploratory forecasting is based upon the past and present knowledge and is oriented toward the future one step at a time. Exploratory forecasting starts from the current situation and explores technical development toward the future. The kernel of exploratory forecasting is to predict future technology development according to its current trend. Specific methods include intuition forecasting, panel approach, Delphi method, brainstorming, trend extrapolation, growth curves, analytical methods, regression and curve fitting, substitution technique, and envelope curves. Delphi method is a type of subjective forecast, and curve fitting is a type of objective quantization survey method.

Normative forecasting first sets up the future needs and objectives and then specifies the means to achieve them. Normative forecasting not only points out some future inventions, but also points out the moving procedure from original ideas to actual technology. This forecasting pays more attention to find the foundation structure of the present development trend than to find the trend. This methodology attaches great importance to all sorts of structure and models that can form the new technology. The kernel of normative forecasting is to carry on technology morphological analysis as a system. This forecasting has relevance trees, scenario writing, and morphological analysis.

Xu [4] classified traditional forecasting methods into five types, which are extrapolation method, leading indicator method, cause–effect method, prospect method, and probability method. Up to now, these traditional forecasting methods still have an extensive application. An exercise of large-scale Delphi was carried out first in 1994 and second in 1999 [5].

Recently, many new forecasting methods were used in technological forecasting. Alex and Kim [6] apply science and technology studies in technology forecasting and scenario planning. Sharma et al. [7] used modified idea-writing to conceptualize and generate ideas regarding flexibility in technology forecasting. Modified idea-writing eliminates the need for collection of the group at one place. Interpretive structuring modeling (ISM) is used as the structuring methodology. ISM is a method of identifying and summarizing relationships that exist among specific items that define an issue or a problem. Since the publication of the Bass model in 1969, research on the modeling of the diffusion of innovations has resulted in a body of literature consisting of several dozen articles, books, and other assorted publications. Mahajan [8] gives a review for the research of new product diffusion models in marketing. Dupeng [9] analyzes the foreground of Taiwan Direct Selling market with Bass diffusion model. Armstrong [10] drew upon findings from the diffusion literature to assess the prospects for the diffusion of expert systems in forecasting. Among the most promising techniques for technological forecasting are the so-called biology-related studies [11]. These include the Fisher–Pry model of 1971 [12] and the 1997's Genetic Re-Engineering of Corporations [13].

These methods have their limitation in the application of technological forecasting. Quantitative analysis forecasting method emphasizes the quantity variable excessively and lacks consideration of quality factor, but these factors are an important aspect needing consideration in strategic management. Furthermore, the discontinuity of technology development broke the basis of using current trend to forecast, while using current trend to forecast future just is the basis of applying quantitative methods to forecast. Combining mathematic model with expert's opinion cannot fundamentally offset this problem, because experts' opinion method is inherently a central tendency process, which cannot forecast the discontinuity of technology development.

In order to improve technological forecasting for the discontinuity of technology development, people adopt two new methods: The first is the innovatory method and value analysis; innovatory method is used for distinguishing great technological change and value analysis is used for promoting and appraising advance

gradual technological improvement. The method makes management voluntarily to the technological innovation course. The second method is the prospect method that is used for forecasting probability of producing technological discontinuity. The prospect method is not a pure determinate forecasting method. The method will draw several possible prospects of the future, and appraise the impact of various types of prospects on the enterprise. The management for innovative course is only limited to the analysis and forecasting inside enterprises, and does not consider the external environment condition and rival and potential technological source outside. The prospect method is used mainly for forecasting the external factors that can cause technological change.

Prospect methods have huge improvement in technological discontinuity forecasting. The improvement is that forecaster's subjective judgement and capability are embodied in forecasting. But this method has some inherent defects; these defects have two points, one point is that the lack of design ability makes this method susceptible to the influence of organizer and forecaster, and the other point is that while facing the numerous forecasters, the forecasting data are often difficult to explain and analyze. So, it is necessary for us to revise and improve this method.

Prospect method is inherently a ranking of future technology components, and a subjective and synthesized judgment, which do not assume that the history will recur. Therefore, the forecasting of the discontinuity of technological development can be improved. The comprehensive judged based on the subjective understanding is similar to economics' utility. We could apply the measure of colony's utility to improve the appropriateness and reliability of prospect method, and this makes conjoint analysis a proper option. Conjoint analysis can yet be regarded as a kind of appropriate method.

2 A Summary for Conjoint Analysis

Conjoint analysis was first proposed in 1964 by Luce [14], a mathematical psychologist, and Tukey [14], a statistician. Soon afterwards, Green and Rao [15] introduced conjoint analysis into the marketing field in 1971. Conjoint analysis has become a type of important method to describe consumers' decision about products and services with many attributes. In the 1980s, conjoint analysis obtained extensive approval and application in many fields. Since the 1990s, conjoint analysis is applied deeper and involves a lot of study fields. In the past 30 years, the technologies to estimate the model of conjoint analysis have had a fast development, and conjoint analysis has more and more application in scientific and commercial research (Carroll [16] and Green [17]). In the mid-1990s, some Chinese scholars began the applicability study of conjoint analysis in marketing (Ke [18] and Xu [19]). Some marketing research companies have applied conjoint analysis technique in projects of enterprise consulting too.

He [20] gives a definition of conjoint analysis: Testee will make overall judgment on the testing object, after knowing the result of testee's answer, factor analysis could be used to estimate testee's preference structure. The method is not a

standardized procedure completely. A typical conjoint analysis should include a series of steps, it involve a lot of procedures and not just a single procedure. The objective of conjoint analysis is to determine that whose attributes and levels of those products and services are the most important to the people surveyed.

The basic steps [21] of conjoint analysis are determining the study problem, designing the experiment, collecting data, conducting analysis with a statistical software, evaluating the analysis result, interpreting the analysis result, and simulating market share.

The first step of conjoint analysis is to determine the study problem. The purpose of the study of conjoint analysis is to measure preference and purchase possibility of the people surveyed to some attributes of a certain product and service. For making a purchase decision the people surveyed must consider many kinds of attributes simultaneously and weigh the advantages and disadvantages among these attributes. The characteristic of the products can be described with limited attributes and levels. The attributes chosen should be those which have distinctive influence on testee's preference and decision-making. The number of levels shouldn't be too large, and each level should include approximately equal numbers. When the utility function is not liner to the value of each level, there should be at least three levels. The establishment range of the level should be more extensive than the real market situation, but the extension cannot exceed too much so that the credibility of appraising work is not influenced.

The second step of conjoint analysis is to design the experiment. In designing an experiment, researchers make some forms for the question and answer of the people surveyed, and make some program for the collection of data. There are often three types of methods. One method is bi-factor approach. Testees are first asked to evaluate and rank different combinations of levels of one pair attribute, then the other pair attribute. This method only assesses a pair of attribute each time. Another method is the complete profiles method. The principle of this method is to first reduce the number of factors for comparison through factor designation, then enumerate all important factors' attributes in testing object card and establish a testing object composed of certain common level of each factor attribute. Testee will then rank different combinations of testing object. Another method is the pair-wise tables method. The basic idea of this method is to first provide different attribute combination through factorial design, then pick up some factor attributes to form proposals of different level value, and finally make comparison on combination of proposals.

The third step of conjoint analysis is to collect data. According to the result of the experimental design, printout sheets or cards for survey, where every combination represents a product to be evaluated. We often use the ordering method and the marking method to measure preference and purchase probability of the people surveyed. Using the ordering method we can more accurately reflect the behavior and attitude of the people surveyed in the market. Using the marking method, collected data in a survey is easier to analyze.

The fourth step of conjoint analysis is to make analysis with a statistical software. The analysis method will vary with the data's quality. The methods that we often use have dummy variable regression, logistic model, and variance analysis. The analysis

step usually is that we first estimate each individual's utility function, then classify the people surveyed according to similar degree of their part-worth, and at last make an analysis of each classification.

The fifth step of conjoint analysis is to appraise the analysis result that was made in the previous step. We evaluate a model's quality from three aspects, as goodness of fit, reliability, and stability. We use coefficient of determination of regression analysis to judge fitting precision of model. We use Pearson's correlation coefficient to judge the internal validity of the model; Pearson's correlation coefficient is a correlation coefficient between realistic evaluate score of holdout card and general utility value that is estimated by the model. We use the test-retest method for reliability analysis. For checking the stability of the solution, we divide the sample into several subsamples randomly, and analyze the subsamples accordingly.

The sixth step of conjoint analysis is to interpret the analysis result. The result of conjoint analysis includes relative importance of attributes, utility of each level, total utility of each combination and statistics of model evaluation.

The seventh step of conjoint analysis is to simulate the market share. To obtain the expected value of a product's market share, we should estimate a ratio of product's market share. The ratio is a forecasting proportion of product's purchased times to testee's numbers. We often use the Logit model, the maximum utility model, and the Bradley-Terry-Luce (BTL) model to simulate market share on a certain product. The hypothesis of LOGIT model is that the possibility function chosen by consumers is nonlinear, strictly increasing function of utility. To apply maximum utility model and BIT model, we assume that the possibility function is a liner one of utility.

3 Analysis of a Stimulant Example

The emergence of a new technology depends on the common effects of many factors; these factors have investing demand in a new technology, potential market value of a new technology, realizable difficulty of a new technology. Realizable difficulty means the difficulty to realize a new technology, supporting degree of relative technology to a new technology, and the competitive power of a new technology with original technology. The result of technical forecasting is a comprehensive judgment value that the forecaster makes about combination utility of these factors in different levels. Applying conjoint analysis, we can make an analysis of colony utility to numerous experts' judgments, and get the result most possibly of a new technology considering the above factors. After comparing the forecasting of new technology with the realistic condition further, we can make a forecasting of the technical development process.

In this chapter, factor one to factor five represent, respectively, investing demand in a new technology, potential market value of a new technology, realizable difficulty of a new technology, supporting degree of relative technology to a new technology, and the competitive power of a new technology with original technology. The different levels of these factors are shown in Table 1. We order the grade of data from easy to difficult. According to the levels of these five pieces of attributes, the number of

Table 1 Attributes and levels of the technology forecasting in studied

Attribute	Level
Investing demand	2.99, 3.99, 4.99 9
Marketing value	32, 48, 64
Realizable difficulty	1, 2, 3
Supporting degree	1, 2
Competitive power	1, 2, 3

The unit of investing demand and marketing value is a million yuan

combination of attributes is 162. The number is a product of the level number of each attribute. Using orthogonal design, we select 26 types of combinations from all these combinations as a forecasting program, and then make the forecasting analysis. This course can be realized in the course of ORTHOPLAN [22] of SPSS software. We generate a ranking of 40 experts’ opinion on 26 combinations through the random number generator of EXCEL and list the combinations and ranking result of the first five experts in Table 2. We list these forecasting program and judgment order in Table 2.

Table 2 Combination program and evaluation order of five forecaster

NO	1	2	3	4	5	6	7	8	9
TZ	2.99	3.99	2.99	4.99	2.99	4.99	4.99	3.99	2.99
JZ	64	48	48	64	48	64	64	32	64
ND	1	2	1	3	3	1	2	3	3
ZC	1	1	1	1	2	1	2	2	1
JZL	2	2	3	3	1	2	3	3	1
STA	0	0	0	0	0	1	1	1	1
Sub1	2	8	1	9	2	5	5	7	9
Sub2	9	7	1	9	6	2	9	6	8
Sub3	8	7	2	3	6	4	8	6	9
Sub4	9	6	9	1	8	5	3	5	5
Sub5	7	7	8	3	3	5	3	7	1
NO	10	11	12	13	14	15	16	17	18
TZ	3.99	4.99	3.99	4.99	2.99	4.99	2.99	2.99	2.99
JZ	32	48	64	32	32	64	32	64	32
ND	3	1	1	2	1	3	1	2	3
ZC	1	2	2	1	1	2	1	2	2
JZL	1	1	1	1	1	1	1	1	2
STA	0	0	0	0	0	1	0	0	0
Sub1	7	9	3	1	5	1	2	5	1
Sub2	7	2	8	3	6	9	1	6	4
Sub3	5	5	4	6	4	9	2	4	3
Sub4	1	4	9	4	7	3	8	5	7
Sub5	6	2	7	5	5	4	5	8	4
NO	19	20	21	22	23	24	25	26	
TZ	2.99	3.99	4.99	2.99	2.99	2.99	2.99	2.99	

Table 2 (continued)

NO	1	2	3	4	5	6	7	8	9
JZ	32	32	32	64	48	48	48	48	
ND	2	1	1	1	2	2	3	2	
ZC	2	2	2	2	2	3	3	3	
JZL	3	3	2	1	2	2	2	1	
STA	0	0	0	2	2	2	2	2	
Sub1	7	5	4	0	0	0	0	0	
Sub2	5	6	8	0	0	0	0	0	
Sub3	5	6	1	0	0	0	0	0	
Sub4	8	8	5	0	0	0	0	0	
Sub5	5	2	1	0	0	0	0	0	

In Table 2, TZ means investing demand in a new technology, JZ means potential market value of a new technology, ND means realizable difficulty of a new technology, ZC means supporting degree of relative technology to a new technology, and JZL means the competitive power of a new technology with original technology. STA represents the characteristics of each proposal; proposals with STA value of 0 and 1 are both ranked by experts, but only those with STA value of 0 are used in model estimation, and proposals with STA value of 1 are used to test the effectiveness of the model, and those with STA value of 2 are used to forecast the change of technology development after these proposals are considered. SUBX represents a judgment order to program by the different forecasters. In evaluation, we select Likert–Nine-Scale. When the grade of a program is 1, this means that the emergence of the program is most impossible from the viewpoint of the forecaster. When the grade of a program is 9, this means that the emergence the program is most possible from the viewpoint of the forecaster.

We can use conjoint course of SPSS software to analyze colony utility of the forecaster’s judgment. Conjoint course in SPSS cannot be finished by using menu at present. For finishing conjoint course, we need to run a procedure. The basic procedure of conjoint course of SPSS is as follows:

```
conjoint plan = ' e:\paper\ technology forecasting data 3.sav '
/data = ' e:\paper\ technology forecasting order cards. sav'
/factors =tz jz nd zc jzl
/subject =subj
/rank =rank1 to rank21
/print =summary only.
SUBFILE SUMMARY
```

Using conjoint, we gather the analysis result of colony utility in Table 3.

Table. 3 Conjoint analysis result of colony utility

Attribute	Average importance	Level	Utility
Investing demand	24.50	2.99	−0.0667
		3.99	−0.0573
		4.99	0.1240
Marketing value	24.23	320	0.0625
		480	0.0625
		640	−0.1250
Realizable difficulty	19.97	1	0.0167
		2	−0.0302
		3	0.0135
Supporting degree	12.14	1	−0.0469
		2	0.0469
Competitive power	19.16	1	−0.0875
		2	0.0625
		3	0.0250
2.1531		Constant	
Pearson’s R = 0.649		Significance =0.0033	
Kendall’s Tau = 0.532		Significance =0.0025	
Kendall’s Tau = 0.532 for 5 Holdouts		Significance =0.2242	

4 Conclusion

In Table 3 [23], the average importance shows the order of importance of the attribute from the viewpoint of the people surveyed. Utility is a utility coefficient that is a utility value of every level of every attribute and shows the order result of every level of every attribute. Smaller (more negative) utility coefficient represents higher ranking of the proposal. Pearson’s R is Pearson correlation coefficient and Kendall’s tau is Kendall grade correlation coefficient. These correlation coefficients are composed of two parts: one is the correlation coefficient of estimated preference value and actual preference value, and the other is the result of significance test of correlation. If the model has a high fitting precision, the correlation coefficient should have a large coefficient. Significance shows a statistical efficiency of correlation test. If the value of significance is smaller than 0.05, the correlation test has statistical efficiency. Constant means the order importance of constant as we assign linear model to utility pattern. Kendall’s tau for five holdouts is used to test internal validity of the model. The result of Kendall’s tau for five holdouts shows reliability of conjoint analysis. The procedure of this test is to first substitute testing cards into conjoint analysis and to obtain forecast value, then to calculate correlation coefficient of forecasting value and actual value, and finally to evaluate the model’s internal effectiveness through correlation coefficient.

From these results, we can receive the following information:
As to the whole sample, we obtain an order of attribute that is successively investing demand in a new technology, potential market value of a new technology, realizable difficulty of a new technology, the competitive power of a new

technology with original technology, and supporting degree of relative technology to a new technology. The order shows the importance of the attribute that affects the preference of the people surveyed to new technology development.

As to the whole sample, the finest combination is such a combination that investing demand is least, potential market value is largest, realizable difficulty is of moderate degree, new technology has a support of relative technology, and the competitive power of a new technology with original technology is powerful.

Pearson correlation coefficient is 0.649, Kendall correlation coefficient is 0.532, and the test is statistically significant. This result shows that the model performs well in fitting and precisely reflects the preference of testee colony.

The correlation coefficient of internal effectiveness test is 0.532, but not statistically significant; this may arise because the sample size of simulation is not big enough. If we have a larger sample and an accurate model, statistical test will be significant.

To summarize, the quantitative approach through conjoint analysis leads to a result consistent with that of theoretical deducing.

In the practical application of technology forecasting, colony preference can be used to reflect the future development prospect of technology more accurately. Because it not only overcomes the deviation that individual forecastings are prone to but also considers the discontinuity of technology development well through subjective judgment. Conjoint analysis has the capability to measure colony preference accurately, so it is a kind of useful method in technology forecasting.

Acknowledgements Sponsored by the foundation of natural science research of China (No 70873079) and Sponsored by the soft science research program of Shanxi Province (No 2008041001-03)

References

1. Xu, Q. (2000) Management of Research, Development and Technology-Based Innovation. Beijing: Higher Education Press, pp. 251–252 (In Chinese).
2. Worlton, J. (1998) Some patterns of technological change in high performance computers. In: Proceedings of Supercomputing. Dorland, FL: IEEE Computing Society Press, pp. 312–320. November 14–18.
3. Bahl S. K., and Arora, K. L. (1989) Techniques of technological forecasting. Defence Science Journal 39(3), 277–285.
4. Xu, Q. (2000) Management of Research, Development and Technology-Based Innovation. Beijing: Higher Education Press. pp. 252–277 (In Chinese).
5. Taeyoung, S. (2008) Technology forecasting and s&t planning: Korean experience. //ftp. mct.gov.br/cct/prospector/Eventos/palestras/TaeyoungShin.PDF.
6. Alex, S., and Pang, K. (2007) Applying science and technology studies in technology forecasting and scenario planning. <http://www.relevanthistory.com/sys-at-work.html>.
7. Sharma, C., Gupta, A. D., and Sushil (1999) Flexibility in technology forecasting, planning and implementation: A two phase idea management study. Management of engineering and technology. Technology and Innovational Conference. PICMET'99. Portland International Conference. Vol. 1, p. 239.
8. Mahajan, V., Eitan, M., and Frank, M. B. (1990) New product diffusion models in marketing: A review and directions for research. Journal of Marketing 54(1), 1–26.

9. Du, P. Study on the forecast Taiwan distributors by Bass diffusion model. www.dsrc.nsysu.edu.tw/dsrc-tn/research/chinese/papers2211/k-1.pdf.
10. Armstrong, J. S., and Thomas, Y. (2001) Potential diffusion of expert systems in forecasting. *Technological Forecasting and Social Change* 67, 93–103.
11. Modis, T. (1999) A second lease on life for technological forecasting. *Technological Forecasting and Social Change* 62, 29–32.
12. Fisher, J. C., and Pry, R. H. (1971) A simple substitution model of technological change. *Technological Forecasting and Social Change* 3, 75–88.
13. Theodore, M. (1997) Genetic re-engineering of corporations. *Technological Forecasting and Social Change* 56, 107–118.
14. Luce, R.D., and Tukey, J. W. (1964) Simultaneous conjoint measurement: A new type of fundamental measurement. *Journal of Mathematical Psychology* 1, 1–27.
15. Green, P.E., and Rao, V. R. (1971) Conjoint measurement for quantifying judgmental data. *Journal of Marketing Research* 8(3), 355–363.
16. Carroll, J. D., and Green, P. E. (1995) Psychometric methods in marketing research: Conjoint analysis. *Journal of Marketing Research* 32(4), 385–391.
17. Green, P. E., Krieger, A. M., and Jerry, W. Y. (2001) Thirty years of conjoint analysis: Reflections and prospects. *Interface* 31(3), 56–73.
18. Ke, H., and Falc, P. A. (1994) Conjoint analysis in marketing study. *Application of Statistics and Management* 13(6), 56–65 (In Chinese).
19. Xu, Z., Fang, T., and Su, W. (2004) Application of conjoint analysis in customers' preferences for attributes of products/services. *Quantitative & Technical Economics* 21(11), 138–145 (In Chinese).
20. He, X. (2003) Multiple Statistical Analyzes. Beijing: China Statistics Press, pp. 304–311 (In Chinese).
21. Miao, J. (2002) Conjoint analysis: A new method for market research. *Journal of Shanxi Institute of Economic Management* 10(1), 25–27 (In Chinese).
22. Zhang, W. (2002) SPSS-11 Statistical Analysis Study Course [M]. Beijing: Beijing Hope Electronic Press, pp. 234–246 (In Chinese).
23. Ke, H., and Ding, L. (2000) Market Survey and Analyzing. Beijing: China Statistics Press, pp. 396–407 (In Chinese).

Towards a New Concept for Supporting Needy Children in Developing Countries – ICT Centres Integrated with Social Rehabilitation

Anders G. Nilsson and Thérèse H. Nilsson

Abstract The presented concept for supporting needy children is based on experiences from our site visits to many children homes, schools and rehabilitation centres in various developing countries around the world. We have analysed the Swedish Model for social rehabilitation of children integrated with modern ICT centres as professional resources for development assistance. The standpoint taken is to strengthen the poor and exposed minority groups in the society. In this sense, we argue for the need to combine social action and ICT research in order to obtain the full potential of intended ventures in the developing world.

Keywords Needy children · ICT centres · Indoors rehabilitation · Outdoors rehabilitation · Value principles · Global development · Action-oriented research

1 Introduction

There has recently been an intensive debate in the media on how efficiently we use our development funds in various developing countries. Do the intended ventures reach and favour the desired target group? The target group is represented by the people who are directly exposed to very hard circumstances and great strains in their daily life such as street children, prostituted girls and boys, lonely mothers and poor families for supporting these marginalised minority groups [7, 11].

It is a great challenge to guarantee that our development funds will get through and support the target group to the greatest possible extent – as close as we could to 100%. We know that lots of funding disappears in local administration and political corruption in too big proportions. A constructive solution is to let researchers take over the development assistance programmes in a larger extent. Researchers are experienced in running bigger and more complex projects as well as being responsible so that the undertaken commissions are performed as planned. Research projects

A.G. Nilsson (✉)

Department of Information Systems, Karlstad University, Karlstad, Sweden
e-mail: anders.nilsson@kau.se

are quality checked and evaluated carefully during the whole process in order to guarantee results and effects of the relief actions for the target group.

The purpose of this chapter is to illuminate how development assistance initiatives can be anchored in academic research. Our research question is as follows:

How can we develop a workable and sustainable concept for supporting needy children in developing countries using the idea of ICT centers integrated with programs for social rehabilitation? (ICT = Information and Communication Technology)

2 The Concept – Model Project Trust

Our vision is to create a Children Centre with rehabilitation possibilities for needy children who are suffering from physical handicaps and emotional disturbances. This model project will at the edge develop the needy children to be more self-dependent persons and give them a raised human dignity in our society. We will create confidence for the children and give a hope for their future – we have therefore labelled our model project “Trust”.

The concept behind our model project is summarised in Fig. 1. Our Needy Children Centre is devoted to three essential target groups and therefore consists of three main co-operating units: Mothers Care Center, Disabled Children Centre

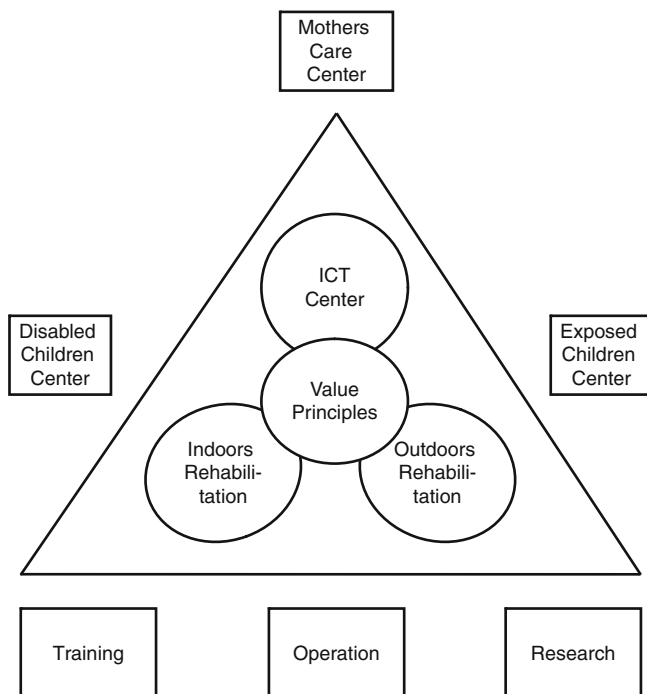


Fig. 1 Concept for Model Project Trust – Needy Children Centre

and Exposed Children Centre. The operations of the Needy Children Centre will be based on six important value principles for rehabilitation services.

The business at the Needy Children Centre will be divided into three main activity areas: ICT centre, Indoors Rehabilitation and Outdoors Rehabilitation. We have not yet found any rehabilitation centre for children running activities in all three areas. We think that our concept is distinctive in the sense that we are taking a holistic view on rehabilitation services. We are convinced that needy children are longing for rehabilitation activities both indoors and outdoors, and that this atmosphere will give a more efficient care. The activities in the ICT centre are indeed an indoor operation. But we will highlight the ICT centre as a separate activity area. The reason for this is that ICT centre exercises are not frequently occurring in rehabilitation centres of today. We regard the direction and scope for the ICT centre as a unique concept for rehabilitation services. Therefore we will focus much effort on establishing a professional ICT centre with modern tools and interesting applications for needy children.

Running a child centre needs a professional infrastructure. We will highlight three essential parts of a good infrastructure: Training facilities, Operation support and Research programmes. Our aim is to have a comprehensive and professional infrastructure. We will establish partnerships with experienced and well-recognised people from different areas of competence. The Needy Children Centre should be placed outside the city district, preferably in the mountains with beautiful natural scenery based on ecological values and principles. We believe that it is important to create a genuine home environment in the countryside with fresh air, clean water, nice woodland, animals and meadows which altogether are good for the mothers and children and their recovery of health [8].

We have since 2002 made site visits to over hundreds of child centres in different countries to gain valuable experiences of modern care and rehabilitation services. We have also visited eminent organisations for children rehabilitation and recreation in order to learn the line of thought behind the Swedish Model for professional child care. These experiences have been a solid base for building up our concept for the Model Project Trust.

2.1 Value Principles

We gained from our site visits a good insight in local conditions for professional care for homeless mothers with babies and for needy children (disabled, exposed). We learnt some valuable lessons which are briefly expressed below:

- Regard all *children as individual subjects* compared to earlier as objects for institutional care in the former planned economies.
- Try to *reunite families* through active support to solve the parents' fundamental problems such as poverty, ill-health, unemployment and alcohol/drug abuse.
- Give disabled and exposed children a *genuine human dignity* through developing their inherent competence such as artistry, sports and service practising.

- Use more advanced and *modern therapy forms* for rehabilitation of children such as ICT centres, adventure rooms, creative artist rooms, hypo massage and hydro massage.
- Intelligent exploitation of various groups of *volunteers for relief work* (e.g. students, parents, priests and social workers) as supplements to the permanently employed staff.
- Understand the fundamental demands of girls and boys during the whole childhood and adolescence and treat them according to similarities and differences in their *genus (sexual) development*.

These six lessons learnt will represent our view on professional rehabilitation services. They will be our value principles on which our Rehabilitation Centre for Children will operate. Our views of children correspond to intentions and content in the UN Convention of the Rights of the Child (CRC) [6]. These rights of the children have been a point of departure for our model project. The CRC is mainly based on four fundamental principles as follows:

- Equal value and rights of all children, prohibition of discrimination (article 2).
- The best interests of the child, a primary consideration in all decision-making (article 3).
- The right of the child to life, survival and development to the maximum extent of the available resources (articles 4 and 6).
- The right to freely express views and to be heard (article 12).

We are convinced that professional child centres of today have to take these legal rights for children into serious consideration and put them consciously into action in their daily life.

2.2 Target Groups

Our Needy Children Centre is devoted to three essential target groups and therefore consists of three main co-operating units: Mothers Care Centre, Disabled Children Centre and Exposed Children Centre. We think that there will exist strong synergetic effects between the three organisational units. Some examples would be as follows. The former homeless mothers could be a valuable resource for taking care of (except their own children) some other disabled and exposed children as volunteers during some hours everyday. It is important that children could learn experiences from their friends from other backgrounds and environments. Exposed children could learn how it is to have some physical handicaps as a disabled person. On the other hand, disabled children could learn how it is to have a hard, severe and poor background as an exposed person. In conclusion, we see great potentials for mutual cooperation between the three organisational units – each representing a main target group for the operations at a children centre.

2.2.1 Mothers Care Centre (MCC)

There is a great need to take care of homeless mothers with their needy children coming from a non-supportive environment. These mothers could join the centre, for example, after a separation (divorce) from their husband or from a life in a severe criminal background or after being a sacrifice for sexual harassment, prostitution and trafficking. The mission for the Mothers Care Centre is to give help to self-help so that the mothers can learn and experience how they can take professional care of their own babies and young children [5]. They will meet other mothers with the same problem background, so they can have a mutual understanding and sympathy for their situation. Mothers with disabled children will also get support and assistance within the other specialised unit in the Needy Children Centre.

A great challenge is to save the life of many young girls who are exposed to prostitution and trafficking. They often miss identity cards and therefore do not formally exist in the society and are without rights to education and health services. These girls get babies that they have to give away – then turn back to the streets again for getting more babies! We need to help these lonely mothers from being exploited and to break the “vicious” circle of unwanted child births in the street. Instead, we wish to give them back a rich life with dignity and respect. We have therefore developed the concept of Mothers Care Centre. This is a professional centre for taking care of a mother who is getting a child as a youngster. The mothers and their children will live in a collective family and manage the daily chores and duties together. We want to construct Mothers Care Centres with the aim of:

- stopping the chain of birth of innocent children in the streets through preventing prostitution, trafficking and sexual abuse;
- giving lonely mothers a home and the opportunity to take care of their children by themselves without having to abandon them and return to the streets;
- creating a genuine family atmosphere so that the mothers and children feel that they are needed and they can be of value for the society and for developing themselves into respectful and decent citizens;
- giving mothers the possibilities to earn their own living in various ways, for example, by cultivating organic or ecological foods;
- giving mothers education and training for life in order to be self-dependent human beings and for having a new life with identity, dignity and respect and
- providing the mothers and children with a meaningful occupation which renders them a new happiness for life.

The location of the Mothers Care Centre should be outside the city centre. This is very essential for keeping the lonely mothers and young girls away from the city streets and preventing them from going back to prostitution and sexual exploitation. In this way a Mothers Care Centre can break the “vicious circle” of unwanted child births in the street. For the recruitment of young girls with babies to the Mothers Care Centre in the countryside we could have an existing children home centre in the city area as a base.

2.2.2 Disabled Children Centre (DCC)

Disabled children have a hard situation living in a more or less isolated environment. Families with disabled children feel a great need to get professional assistance and support for taking care of the physical handicaps of their children. The mission of the Disabled Children Centre is to offer a rich selection of modern rehabilitation forms, indoors as well as outdoors exercises, in order to strengthen and recover the health of the disabled children. We will focus the rehabilitation treatment mainly on children with some cognitive disturbance, mental retardness, speech impairment, autism, intellectual difficulties in learning and limitations in their movement ability. Families with disabled children will meet each other in the child centre and could exchange valuable experiences. Also, the disabled children have possibilities to find playfellows with similar interests to make a break from their otherwise isolated existence.

2.2.3 Exposed Children Centre (ECC)

Exposed children have much in common that they come from a severe background where they have lost the confidence in grown-up or adult people. They could come from various environments and they could be street children, sexually abused children, poor children, orphan children, homeless children, etc. The mission of the Exposed Children Centre is to give such children a home with family atmosphere, based on tenderness, cordiality and security. It is important to let the exposed children have a complete and harmonious development “heart to heart” in a social environment of happiness, love and understanding. We want to improve conditions for children in jeopardy and to create opportunities for the best beginning in their new lives [13]. The children will participate in the planning and decision-making which affect their lives.

2.3 ICT Centre

Children in general are very excited and amused by working with computers! This is the main argument for why we are so eager to set up a special ICT centre for needy children. When children use computers they have a possibility to develop both cognitive and mobility competence. The child can experiment, acquire new experiences, obtain stimulation, make independent choices, develop a power of initiative, become confident, increase innovativeness, practice communication, balance, mobility and concentration as well as learn how to handle mistakes.

This ICT centre in our Needy Children Centre will be organised in two separate sections: (1) Multimedia Studio called Computer Play Centre and (2) Test Studio called Ozlab. There are also other possibilities, e.g. mobile phone laboratory of image talk technology, smart-home technology for disabled children, computer support for creating art/music or using the “One Laptop per Child” (OLPC) approach.

2.3.1 Multimedia Studio – Computer Play Centre

The Multimedia Studio, called Computer Play Centre, will offer paedagogical software for training and learning purposes [2]. This studio will be jointly used by all children in our centre: mothers' children, exposed children and disabled children. For disabled children with cognitive dysfunctions the primary aim is to raise their power of concentration and ability to communicate with their environment. For all children the multimedia products give an arena for playing activities. There exist at least 50 different multimedia programs from serious vendors on the market today. Most of the products have a twofold goal to support the children's need for educational and playing activities.

Computer games is a method of making play possible and fun while developing the abilities and independence of the child or young person. Before being in a position to use the multimedia studio, there is a need to translate a selection of the existing paedagogical software on the market into the mother tongue language, from e.g. English source languages. The optimum size of the multimedia studio is around ten fully equipped computers. Besides multimedia applications there should be installed programs for office products and Internet applications. A possibility is to use special office products for disabled children with speech-based facilities.

2.3.2 Test Studio – Ozlab

The Test Studio will be specially designed for disabled children with cognitive disturbances. This studio, called Ozlab, is based on Wizard-of-Oz technique. In Wizard-of-Oz experiments, a test person (in this case a disabled child) thinks he writes and speaks to the computer in front of him when in actual fact the test manager sits in the next room interpreting the user's commands and providing the right answers [9, 10]. The Ozlab experiment environment is a prototype software worked out by a professional team of researchers from linguistics, special education and computer science. The Ozlab concept could be used for many different applications but have up till now mainly been used for math exercises. The test studio Ozlab requires two adjacent rooms with a mirror wall separating the test person (disabled child) from the test manager (occupational therapist). The Ozlab studio also requires equipment of two specially designed computers in a network together with an ordinary digital camera for video recording of the test situations.

2.4 Indoors Rehabilitation

The Needy Children Centre should have several professional facilities for indoors rehabilitation of disabled and other needy children. This means the use of more advanced and modern therapy forms for rehabilitation services. We have mentioned above the need for building a well-equipped ICT centre. We will here mention some other types of necessary indoors activities for efficient rehabilitation of children. The indoors rehabilitation from our Needy Children Centre will be organised in

three separate sections: (1) Adventure Rooms, (2) Creative Artist Rooms, and (3) Physical Training and Massage Rooms.

2.4.1 Adventure Rooms

The Adventure Rooms (or Multi-Sensory Rooms) represent a specially designed environment where the children can have reinforced sense stimulation under safe conditions. The purpose is to counteract that the children will become self-absorbed or introverted and so to say that they live in their own world. The experiences in an adventure room should give a feeling of rich pleasure and be highly developing for the children. The idea behind the adventure concept is to focus on one of the child's human senses at a certain time. We need a set of rooms for different adventure and sense experiences. In a music room will the sense of hearing be in the centre of interest. Being in a waterbed you can feel the musical vibrations from the loudspeakers. You can play on relaxing African drums and on a soft synth. In the light room we have immense possibilities to various visual impressions. The water-filled bubble pipes shine in all their colours and the fibre optics gleams beside the waterbed. In the touching room the senses of feeling are in the foreground. You can touch on spiny brushes, soft skins and vibrating play things. Here you can experience a huge "ball bath" with thousands of coloured balls.

2.4.2 Creative Artist Rooms

The Creative Artist Rooms develop the children's artistic talents in several possible directions. We need separate rooms for different artistic activities such as painting, making handicrafts and performing as drama actors on stage. Both disabled and exposed children have an inherent potential for artistry which they seldom have the opportunity to develop in a professional way. Therefore, it is a challenge for them to show their utmost capabilities and talents in a supportive environment within the child centre.

2.4.3 Physical Training and Massage Rooms

The Physical Training and Massage Rooms are a "must" in rehabilitation centres for needy children. In this environment, we have various therapy forms for gym activities, body massage, hydro massage (water massage), mobility exercises in water pools, etc. It is important to make individual plans for the children about their needs for special treatments.

2.5 Outdoors Rehabilitation

Placing the Needy Children Centre in the countryside gives a good opportunity for outdoors recreation. Different excursions to the mountains, forests and lakes will be

done on a regular basis. Also, special outdoors arrangements for playing and sports exercises will be organised. For disabled children we will have activities for hypo massage with horse riding. The environment for the children will be supportive for getting acquainted with many animals in the countryside – therefore a small animal farm will be connected to the child centre.

A unique element in our concept is to run a special activity or project labelled “Sun, Water and Wind”. The purpose is to create prerequisites for an active outdoor life in the nature with possibilities for training, recreation and plantation exercises. We propose a specially designed construction consisting of three parts:

- *The Forest Walker’s Path*: A sense training way through the woodland containing a number of stations aiming to stimulate different senses: reflections, sounds, and thrilling things to touch and smell.
- *The Adventure Land*: Training track with different actions for physical training. The adventure land starts in the woodland and ends up at a lake shore. Balance, motor activity, and physical strength are exercised on different surfaces, sloping grounds and in climbing nets, ladders and scaffolds. Clever and funny solutions will stimulate cooperation and common trouble-shooting.
- *The Garden Plot*: This place smells, tastes and gives a delight for the eye. Plantation beds will be adapted so that everyone can get a place to set, sow and reap. The plot should be the Garden of Possibility!

Keywords for this special activity or project are that the construction should be developable, be long-term based, reach the disabled as well as other needy children, have a good accessibility and last but not least be stimulating to all human senses.

2.6 Infrastructure

Running a child centre needs a professional infrastructure. We will highlight three essential parts of a good infrastructure: Training, Operation and Research.

2.6.1 Training

Competence development is a key factor to be successful in the daily operations of a child centre. Our concept introduces a lot of new ideas, knowledge and experiences from different subjects such as psychology, education, sociology, medicine, agriculture, nutrition and information systems (IT). We need to have various courses for the staff, mothers and families (of disabled children) to raise their competence and give them up-to-date information on important issues. The training facilities should be located at the Needy Children Centre and performed by professional teachers from areas of vital importance for the business operations.

2.6.2 Operation

There is a need for a good infrastructure to support the daily operations at the Needy Children Centre. We have to plan for operations for children living permanently at the centre (mothers care centre, exposed children), children staying for a couple of weeks (summer camps, longer rehabilitation) and children coming over the day (shorter rehabilitation). The staff at the Needy Children Centre will be a combination of employed people and volunteers. The rehabilitation of the children will be done in a natural way in order to fulfil their fundamental needs of nutritious food, good hygiene, necessary health care and purposeful education. A rehabilitation centre for needy children according to our concept requires a flexible organisation with a rather high degree of freedom for the staff. On a regular basis we will make a quality assurance of the operations at the Needy Children Centre.

2.6.3 Research

The rehabilitation services at the Needy Children Centre should be based on up-to-date knowledge from recent research studies in important areas and subjects for the daily operations. Most essential is to gain professional competence and skills in child development, special education and physiology. For the ICT centre exercises, it is important to base on frontline research on, for example, multimedia applications from disciplines such as information systems science, computer science and media & communication science. Research-based studies could be used for assessment, evaluation and follow-ups of the business at the Needy Children Centre. We will be engaged in comparative studies of rehabilitation services for needy and disabled children between some selected countries.

3 Actions and Research in Concert

Our mission is to promote social change and relief work of needy children on a scientific basis. We focus on supporting needy children and their mothers with urgent relief actions. The purpose is to develop and apply new concepts in order to stimulate the children's ability to communicate and concentrate in school. This takes into account different programmes for supporting education, recreation and rehabilitation of the children. An important loadstar is that scientific knowledge from psychology, paedagogy, social work, as well as IT and media is launched and implemented globally in various developing countries. The projects are based on knowledge of mental thinking, learning, creativity, stress and health.

Our Model Project Trust will use a well-known approach labelled "Base of Pyramid" which is perfectly timed for developing countries [3]. This means that the new economy of the country is growing through developing the poor population to be self-dependent, productive and respectful citizens. Such a global development is performed through combining social programmes with environmental and economic development of the country (see Fig. 2). According to a "Base of Pyramid"

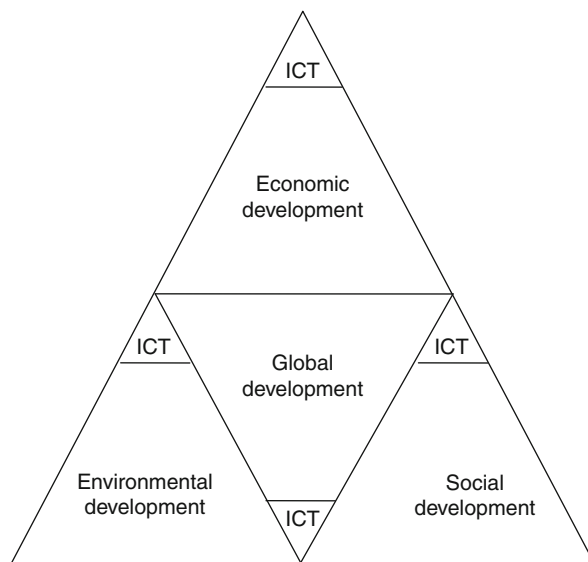


Fig. 2 Fostering global development with ICT using a “Base of Pyramid” approach

approach, all development measures in a future modern civil society must be more integrated with ICT. We have focussed on ICT centres integrated with social rehabilitation of children and mothers as a part of a country’s global development.

Therefore, our project is important for supporting needy children and their exposed mothers to give them a new career and progress in life. The progress of the child and mother will be measured through an initial investigation of the basic needs and by regular follow-up studies of, e.g. cognitive development and health improvement. This kind of project could be a valuable model and a good image for multiplying investments in professional children centres on a universal scale in developing countries around the world.

Realised projects can act as illustrative examples and prototypes for continued ventures in developing countries. The research will be done as an integrated work between concept development (theory) and field work at site (practice). The interplay between academic knowledge and development assistance is established with mutual benefits. Actions and research are in concert. This research method is characterised as “consumable research” [4, 12] based on theoretical knowledge integrated with business practice from the ICT and development field.

A researcher has a great capacity for analysing problem situations, finding constructive solutions and creating sustainable results. Researchers have an excellent competence to design systems with self-realising effects. The system solution is about breaking vicious circles and hence fostering good circles in our society. What really counts is how we can reach the roots and causes behind the social problems and to keep away from always trying to relieve the great suffering and distress in

the society, which of course is an indication of serious symptoms. The leading principle is to do the right things from the very beginning instead of dealing at the end with all system faults in the society. The challenge is to create new prerequisites and to remove the obstacles for a desired progress in society. This is called a second-order change [14] or a double-loop learning [1]. We have in our presented concept launched the idea of implementing Mothers Care Centres (MCC) in order to break the vicious cycle of prostitution and unwanted child births in the streets.

Action-oriented researchers have a good capability to manage challenges connected to “Base of Pyramid”. Therefore, let researchers take over and carry out development assistance projects seriously in different developing countries. We have in this chapter illustrated a new and innovative concept for supporting needy children in developing countries using ICT centres integrated with programmes for social rehabilitation.

References

1. Argyris, C., & Schön, D. A. (1978) *Organizational Learning: A Theory of Action Perspective*. Reading, MA: Addison-Wesley.
2. Burnett, R., Brunstrom, A., & Nilsson, A. G. (eds) (2003) *Perspectives on Multimedia: Communication, Media and Information Technology*. London: Wiley.
3. Hart, S. L. (2007, 2/E) *Capitalism at the Crossroads: Aligning Business, Earth, and Humanity*. Upper Saddle River, NJ: Pearson Education and Wharton School Publishing.
4. Håkangård, S., & Nilsson, A. G. (2001) Consumable Research in Information Systems: Perspectives and Distinctive Areas. In: Nilsson, A. G., & Pettersson, J. S. (eds) *On Methods for Systems Development in Professional Organisations: The Karlstad University Approach to Information Systems and Its Role in Society*, pp. 7–31. Lund, Sweden: Studentlitteratur.
5. Holmes, J. (1993) *John Bowlby and Attachment Theory*. London: Routledge.
6. James, A., Jenks, C., & Prout, A. (1998). *Theorizing Childhood*. Cambridge: Polity Press.
7. Lewin, K. (1947) Frontiers in Group Dynamics – Part II. Channels of Group Life, Social Planning and Action Research. *Human Relations* 18(2): 143–153.
8. Lindstrand, A., Bergström, S., Rosling, H., Rubenson, B., Stenson, B., & Tylleskär, T. (2006) *Global Health: An Introductory Textbook*. Lund, Sweden: Studentlitteratur.
9. Nilsson, J., & Siponen, J. (2005) Challenging the HCI Concept of Fidelity by Positioning Ozlab Prototypes. In: Nilsson, A. G., Wojtkowski, W., Wojtkowski, W. G., Wrycza, S., & Zupancic, J. (eds) (2006) *Advances in Information Systems Development: Bridging the Gap between Academia and Industry, Proceedings of 14th International Conference on Information Systems Development, ISD'2005*, Karlstad, Sweden, pp. 349–360. New York: Springer.
10. Pettersson, J. S. (2003). Ozlab – A System Overview with an Account of Two Years of Experiences. In: Pettersson, J. S. (ed.) (2003) *HumanIT 2003*. Karlstad University Studies 2003:26, Karlstad, Sweden, pp. 159–185.
11. Rawls, J. (1999) *A Theory of Justice*. Oxford and New York: Oxford University Press.
12. Robey, D., & Markus, M. L. (1998) Beyond Rigor and Relevance: Producing Consumable Research about Information Systems. *Information Resources Management Journal* 11(1): 7–15.
13. UNICEF (1999). *Generation in Jeopardy – Children in Central and Eastern Europe and the Former Soviet Union*. Armonk, New York and London: M.E. Sharpe.
14. Watzlawick, P., Weakland, J. H., & Fisch, R. (1974) *Change: Principles of Problem Formation and Problem Resolution*. New York: W.W. Norton & Company.

An Investigation of Agility Issues in Scrum Teams Using Agility Indicators

Minna Pikkarainen and Xiaofeng Wang

Abstract Agile software development methods have emerged and become increasingly popular in recent years; yet the issues encountered by software development teams that strive to achieve agility using agile methods are yet to be explored systematically. Built upon a previous study that has established a set of indicators of agility, this study investigates what issues are manifested in software development teams using agile methods. It is focussed on Scrum teams particularly. In other words, the goal of the chapter is to evaluate Scrum teams using agility indicators and therefore to further validate previously presented agility indicators within the additional cases. A multiple case study research method is employed. The findings of the study reveal that the teams using Scrum do not necessarily achieve agility in terms of team autonomy, sharing, stability and embraced uncertainty. The possible reasons include previous organizational plan-driven culture, resistance towards the Scrum roles and changing resources.

Keywords Agile software development · Scrum · Agility indicator · Autonomous team · Context sharing · Stability · Uncertainty

1 Introduction

Agility is a multifaceted concept and has been interpreted in many different ways in both system development research and practice [5]. It originates from several disciplines including manufacturing, business and management and has root in several inter-related concepts, such as flexibility and leanness. Based on the comparison and contrast of these concepts, Conboy and Fitzgerald provide a broad definition of agility as “the continual readiness of an entity to rapidly or inherently, proactively

M. Pikkarainen (✉)
VTT, Technical Research Centre of Finland, Espoo, Finland
e-mail: minna.pikkarainen@vtt.fi

or reactively, embrace change, through high quality, simplistic, economical components and relationships with its environment” [6], p. 40. In Information Systems Development (ISD) context, ISD agility is concerned with why and how ISD organizations sense and respond swiftly as they develop and maintain information system applications [12].

In software development domain specifically, although agile software development methods, such as eXtreme Programming (XP) [2] and Scrum [21], have emerged and become increasingly popular in the past decade, the meaning of agility is yet to be fully understood in this domain. Wang and Conboy [22] identify a set of agility indicators through investigating software development teams using agile methods, but the conclusion they have drawn is based on the study of XP method only. At the same time, as the teams using Scrum are not yet evaluated from agility perspective, the generalizability of these indicators to other agile methods has yet to be validated.

Based on this observation, this study sets out to investigate the meaning of agility in software development teams using Scrum development method, utilizing the indicators developed by Wang and Conboy [22]. Scrum has been pioneered by Schwaber and Beedle [21] and is one of the most popular agile methods adopted in many companies. It was originally influenced by Boehm’s “spiral” model, but it was developed based on industrial experiences to simplify the complexity of the project and requirements management in software organizations [20]. Scrum describes practices on an iterative, incremental time-boxed process skeleton. At the beginning of the iteration, the team has a sprint planning meeting in which they decide what the team will do during the following iteration. At the end of the iteration, the team presents the results to all the stakeholders in the sprint review meetings to gain feedback on their work. The heart of Scrum is an iteration in which the self-organizing team builds software based on the goals and plans defined in the sprint planning meeting. The team also has a daily 15 min meeting called the daily Scrum, in which they check the status of the project and plan the activities of the next day [20].

The remaining part of the chapter is organized as follows: Section 2 briefly introduces the previous work that leads to this study; Section 3 describes the research method and the context of the empirical study; the findings are presented in Section 4; Section 5 is a discussion of the findings in the light of the relevant studies. A summary section wraps up the chapter with the implications and limitations of the study as well as the future work.

2 Agility Indicators for Software Development Teams

Wang and Conboy [22] identify a set of agility indicators for software development teams. Two facets – autonomous but sharing team and stability with embraced uncertainty – are of particular relevance to this study (see Table 1).

Table 1 Agility indicators (adapted from [22])

Agility facets	Manifested in software development
Autonomous but sharing team	Distributed competences Disciplined team Knowledge sharing Context sharing Collective ownership of results
Stability with embraced uncertainty	Short-term certainty Team being satisfied, motivated and focussed Working at a sustainable pace Probability to change directions Having a whole picture of the project

2.1 Autonomous but Sharing Team

Agile advocates suggest that software development processes should be organized to improve and distribute both technical and social competences continuously [4]. Auvinen et al. [1] discover a competence buildup in a team where several agile practices are piloted. Wang and Conboy [22] suggest that a team composed of autonomous but interconnected developers has a tendency to be agile. In agile teams, competences are not concentrated on few people so that there is no bottleneck in the development process. Team members are confident and courageous in the interactions with customers and with each other. Meanwhile, contrary to the view that agility means chaos, Beck and Boehm [3] and Rakitin [17] argue the importance of discipline in agile processes. An agile team is composed of disciplined, self-responsible and committed individuals. Discipline is an essential component of an autonomous team [22].

Sharing is a common characteristic of agile teams, including both knowledge sharing and context sharing [11, 14, 16, 19]. Melnik and Maurer [14] believe that the so-called “background knowledge” about a project is important to achieve effective communication. It is important for all team members to have a common frame of reference – a common basis of understanding. Poole and Huisman [16] observe that, in the company they have studied, there is a measurable increase in the visibility of what everyone is doing on the team after the adoption of the agile practices. The improvement in visibility is considered one of the greatest successes the company has achieved. Wang and Conboy [22] argue that, to effectively self-manage, a team needs to share the understanding of their working context in addition to knowledge sharing. Context sharing is a precondition to provide effective feedback, interpret them in a sensible way, and take appropriate actions. Sharing also means results sharing, such as collective ownership of code and solutions, which reduces the risk of knowledge loss and increases the sense of being a true team. Fredrick [11] reports the experience of collective ownership of codes. When it is realized, even the most complex business problems can be easily figured out. In contrast, individual ownership of code makes people defensive – people take it personally when someone

suggests their code does not work. Schatz and Abdelshafi [19] also document the collective ownership in their experience report where developers took ownership of the features they created and took pride in showing their work to the stakeholders during sprint reviews. Rising and Janoff [18] notice that in a team they have studied, at every meeting, as small tasks were completed and the team could see progress towards the goal, everyone rejoiced.

2.2 Stability with Embraced Uncertainty

Wang and Conboy [22] emphasize that stability is a desired property of development teams that gives developers a sense of security and control over what they are working on. It can be drawn from a short-term certainty provided by a time-boxed development process. Stability for development also means a team is working at a sustainable pace, focussed and motivated, and working with ease and satisfaction. Several studies have noticed team satisfaction and motivation in agile teams (e.g. [9, 18, 19]). For example, Drobka et al. [9] conduct a survey of a team using XP and find that it creates a surge in morale since XP provides constant feedback to the developers and at the end of each day the team has a working product. Team members gain a sense of accomplishment from their daily work because they could immediately see the positive impact their efforts have on the project. When morale is high, people are excited about their work, leading to a more effective, efficient development team.

Meanwhile, uncertainty is inevitable in software development. It comes from both the environment a team is embedded in and the development process itself. Managing uncertainty does not mean to predict what is going to happen and do future proof work today. It is to ensure the probability to change the direction a team goes towards but meantime not to get short-sighted, to have a whole picture of the project in mind, and to let solutions emerge [22]. It is echoed in [10] which suggests that, when using the XP practices, especially the simple design, one should look ahead and do things incrementally, in order to have a big picture.

3 Research Approach

This study employs a qualitative research approach, treating agility as a qualitative property of a software development team that can be better studied through words and the meanings people ascribe to them. The specific research method used is case study. A multiple-case study design is employed, since the study intends to be a cross-sectional study. The research results would be more convincing when similar findings emerge in different cases and evidence is built up through a family of cases [15]. A software development team is taken as a case, and the level of inquiry is at the team level. Semi-structured interviews are the main data collection

method. Interviews are transcribed verbatim, imported to NVivo and analysed using the agility indicators as analytical lens.

Two companies that were selected for this research were both market leaders of specific products working in dynamic, global market environments. Both companies originally deployed Scrum method because they had a clear need to respond to the needs of the changing market situation and to produce products to the markets faster. Both companies were SMEs that had the key development group in one European country but market offices all over the world.

Case company 1 produces commercial products in information safety domain. The use of Scrum method is integrated to their company-level process model. The team involved in this study, Team A, has four developers, one quality engineer and a Scrum master. Case company 2 provides hardware-driven embedded commercial products. Scrum has been used in some specific project teams for 4 years. Team B, the team this study investigates, is a Scrum team producing a platform product which is strategically important for the company. It consists of five developers and a project manager who is also doing the actual development work.

4 Case Analysis

Using the agility indicators as analytical lenses, the analysis of the two Scrum teams revealed a set of issues faced by the teams when they strive to achieve agility using Scrum.

4.1 Team A

Team A had a difficult start when time boxing did not work for the first three iterations: “We didn’t do very well with the time boxing in the first two or three sprints” (Developer). Developers felt that, due to the use of Scrum method, they needed more testing and integration skills than before: “For me testing was difficult, I do not know it so well, that is a reason. It also was a communication problem” (Developer). Furthermore, it was challenging for developers to take a full responsibility of the project decisions: “I was not so sure, if I dare to take this decision by myself or not” (Developer). As soon as the Scrum framework was fully taken into actual use, however, the team made the “time boxing” working at all levels: “Basically the setup was so that we had two days [of meetings]. The first day we had review meeting, then the next day we had the sprint planning meeting... Every month two days, only for these activities” (Manager). As a consequence, two agile practices, unit testing and continuous integration, started to be used effectively by the team through discussion in iteration retrospective meetings. Meanwhile, both knowledge sharing and context sharing are improved among the team members: “Information was totally shared within the team” (Manager). Over the time, the team moved one step closer to their vision and the developers started to take more responsibility of the decisions

related to product design: “In the end of the project we did all the decision as a team. . . We made all decisions that are related to the design” (Developer). However, knowledge sharing between developers and stakeholders did not seem to be working well: “Everything we got was like second hand information” (Developer).

Team A was good at keeping their short-term certainty at the beginning. The product management and the team made requirements prioritization and analysis during the sprint planning meetings, and all the stakeholders were happy about the results presented in the sprint reviews. However, the situation changed radically when the project started to have increasing amount of features in the product backlog and the Scrum master and product owner lost control over the backlog management: “As I said I run it, the backlog wasn’t properly organized, it wasn’t prioritized” (Manager). The time-boxed meetings were not enough to assure either the long- or short-term certainty: “So most of the time we didn’t know what the feature was, we did not spend a lot of time analyzing the feature” (Developer), and the project lost the short-term goal: “We don’t really have a clear goal” (Manager). Instead of embracing changes and uncertainty in the project, the purpose of the developers was to reduce the amount of changes: “We tried to reduce the risks on the interface side by trying to reduce the amount of changes around the project” (Developer). Furthermore, the developers refused to comment on the technical solutions of the product in sprint planning meetings because they did not have possibilities to communicate with actual customers: “We were never able to talk to the people who might have an impression of how it should behave” (Developer). As a result, the developers were not satisfied with the way they were working. Some developers particularly did not like Scrum meetings: “Some of the developers were not very happy at all with the sprint” (Manager).

4.2 Team B

Team B was deploying Scrum in a quite late phase of the overall development cycle. Due to this late deployment, the developers and the project manager found them suddenly start to participate in sprint planning meetings. Meantime, however, the project manager and the senior management did not want to change their way of communicating. They still had their own weekly status meetings: “Originally those meetings were held to solve conflicts, now they are used in status monitoring” (Manager). Context knowledge about the project was not shared with the whole team due to the separation of meetings. However, although the developers were located in separated rooms, all specification and technology knowledge were well shared between the team members. The strong communication was achieved using workshop techniques: “We talk about all techniques and specifications with the whole product group, the purpose is to get everyone’s opinion” (Developer). However, decision making was not distributed in Team B. In fact, the project manager was still solely responsible in the decision making: “Project manager alone is responsible for all decision making” (Manager).

The developers were happy during the first two increments as long as the manager gave them peaceful time to work towards the increment goals. However, the situation changed during the third increment because of new emergent customer demands. The consequence of the demands was that the team had to suddenly work on a different project: “After second increment, we lost this possibility when other work tasks appeared; the third sprint was terrible” (Manager). At the implementation stage, the management also faced a new challenge. They had difficulties in managing resourcing activities. Thus, the developers had to work in several projects at the same time, changing between projects during the working sprints: “Those resources that were booked for the project have been stolen later on” (Manager). As a consequence the team lost the short-term certainty: “It is not possible to say what you are doing next Thursday, because you never know what emails you have got during the night before!” (Developer). Somehow, the project team managed to make releases after every 2 months, but the goals of the increment could not often be achieved due to the resourcing problems. Meanwhile, the developers found that it was not easy to change project directions due to a lack of tool support: “We do not really have requirements management tool, the tool that we use is more like bug fixing data” (Developer). Change management is complex in the case of Team B also due to the fact that the team uses software product line techniques. For example, feature impact analysis (where features are reported in a product backlog including possible release information and estimations; in the sprint planning, features are divided into smaller working tasks and pointed out for each available resource) turned to be a challenge for both developers and project management in software product line environment. The use of Scrum did not help the team to maintain reusability and to implement commercial off-the-shelf products using technical standards. The issues discovered in the two Scrum teams are summarized in Table 2.

Table 2 Agility issues in the two Scrum teams

Agility facets	Manifested in software development	Agility issues in the Scrum teams	
		Issue	Reason
Autonomous but sharing team	Distributed competences	Communication was difficult about testing competencies Lack of continuous integration competence	Testers had resistance towards the change No previous experience; difficulties to understand the real meaning
	Disciplined team	Resourcing was not working; planned features	Not enough people; customer projects always in the first priority

Table 2 (continued)

Agility facets	Manifested in software development	Agility issues in the Scrum teams	
		Issue	Reason
Stability with embraced uncertainty	Knowledge sharing	Customer information was second hand for developers	Authority of product owner and Scrum master in customer communication
	Context sharing	Difficulties to understand the stakeholders	Previous organization culture
	Collective ownership of results	Difficulties for developers to take responsibility	Previous organization culture
	Short-term certainty	There was no clear short-term goal in the project	There was not enough time to analyse features
		Changes were not analysed deeply enough	Agile way of working did not support change request analysis
	Team being satisfied, motivated and focussed	Some of the developers did not like meetings	Meeting time consuming; lack of preparation; not enough time for feature analysis
		Changing resource situation decreased developers' motivation	No peaceful time to work planned working tasks
	Working at a sustainable pace	No knowledge of what will happen in the next week	Developers were resourced in several projects at the same time; some projects in maintenance mode
	Probability to change directions	Changes in the plans time consuming; avoiding of changes	Product backlog was poorly managed
		Tools do not support management of evolving requirements	Company policy
		Change management was complex	Use of software product line architecture
		No time to analyse features	Poorly organized product backlog; no requirements prioritization before iteration planning meetings

5 Discussion

The analysis reveals that there are several significant agility issues in both Scrum teams. It seems that neither of the two Scrum teams is agile in terms of team autonomy or stability with embraced uncertainty. Case 1 is more agile in terms of short-term certainty and team autonomy but fails to maintain the whole picture of the project and knowledge sharing with customers. Case 2 is able to hold the overall picture quite well, but has no ability to achieve short-term certainty and knowledge sharing between the team members. In both cases, developers are unsatisfied and unmotivated. The reasons behind these issues appear to be (1) previous organizational plan-driven culture; (2) resistance towards the Scrum roles; (3) use of technical standards; (4) reusability goals; (5) lack of tool support; (6) lack of needed agile competence/skills; and (7) evolving resources.

Notwithstanding the implications of current agile development models, methods and frameworks as well as the increasing interest of industry as described in previous research [6, 8, 13, 18], there are still concerns that the use of agile method itself does not answer to the company goals to be flexible, and rapidly produce working software. Analysis provided in this chapter supports an assumption that the Scrum teams do not necessary fulfil the goals of agility in terms of continual readiness and proactively or reactively embrace to change as described in [6]. Although there are several experience reports that describe success stories of the use of Scrum method in information system development team, most often they do not reveal critical issues that the real development teams are dealing with.

Rising and Janoff [18] report on the use of the Scrum method in three small software development teams. Similar to our cases, these teams have (1) difficulties with developers' attitude towards everyday meetings; (2) lack of competence on estimation and continuous activity planning; and (3) complexity of product backlog management. One consequence of the situation is that the Scrum team overcommitted themselves taking too many responsibilities and managers had to modify the product backlog to reflect to the new strategy and company process model which was against the Scrum development.

Derbier [7] reports Solystics experience of the use of Scrum method, focussing especially on the issues revealed among the developers and managers in Scrum development. According to the experiences of large and complex system development, it is shown that from project level it takes often much time from people to really understand the meaning of Scrum meetings. Additionally, the use of Scrum demands new communication skills in the situation in which the individual contribution is easily hidden. Furthermore, estimation process needs to be well shared with developers. From the organizational level, long-term visibility is difficult to manage in a Scrum project.

Mann and Maurer [13] report on the Scrum impacts on customer satisfaction and overtime work in the teams. Based on the empirical analysis of the case study in which Scrum was used in a development project, they reveal that although it is sometimes difficult to follow sprints of 30 days and hold daily Scrum meetings as a Scrum practice, facilitating customers to keep up to date with the development work

and planning meetings help to reduce confusion about what should be developed from the customer perspective. Similar to Mann and Maurer's case, following time-boxed 30-day sprints and holding daily meetings and time-boxed sprint planning meetings turn out to be difficult also for the two cases we have studied.

Schatz and Abdelshafi [19] examine the use of Scrum in Primavera. Based on their experiences they reveal that in a Scrum team, developers have sometimes difficulties to manage increasingly growing bug lists which can cause a situation in which the team produces high amount of features that, however, are not in good enough condition to show to stakeholders in sprint review meetings. Furthermore, developers, in this case, put too much emphasis on stakeholders' comments taking them always into the development sprint. In the analysed cases, from the management perspective, Scrum made it difficult to predict releases and to assure product maintainability in the long term. For example:

- People were worried about the role and responsibility change.
- Developers took stakeholders' comments too easily, although it would not always have been necessary.
- Developers showed features that were not fully tested in sprint review.
- Backlog of bugs was growing; many features were not in good enough condition.
- Losing sight of technical infrastructure and long-term maintainability.
- Scrum made it difficult to determine how far you are from release because of the requirements change.

Dingsoyr et al. [8] report results of action research made for Scrum development team in Avinor. As an issue they have identified problems with estimation and backlog management. The problems with effort estimation, lack of model for the action between the different parties and the lack of time to complete the backlog were the same, defined also in our case study.

6 Conclusion

The use of agile methods has increased dramatically during the past years. However, the meaning of agility is not fully understood by either part of the research communities or ISD enterprises. In this chapter, two Scrum teams were analysed using previously identified agility indicators: (1) autonomous but sharing team and (2) stability with embraced uncertainty. In the future, we continue the analysis of the presented learning aspects of agility in the Scrum teams. Because the use of two cases is not enough to do generalization of the results, we also intend to continue analysis with the additional agile cases. This is done in order to further validate the presented agility indicators.

Acknowledgements This work was supported, in part, by the Science Foundation Ireland grant 03/CE2/I303_1 to Lero – the Irish Software Engineering Research Centre (www.lero.ie) and TEKES to VTT, Technical Research Centre of Finland.

References

1. Auvinen, J., R. Back, J. Heidenberg, P. Hirkman, and L. Milovanov (2006). Software Process Improvement with Agile Practices in a Large Telecom Company. In: *Proceedings of Product-Focused Software Process Improvement*. Springer, Berlin, LNCS 4034, 79–93.
2. Beck, K. (1999). *Extreme Programming Explained*. Addison Wesley, Reading, MA.
3. Beck, K. and B. Boehm (2003). Agility through Discipline: A Debate. *Computer* 36(6), 44–46.
4. Cockburn, A. and J. Highsmith (2001). Agile Software Development: The People Factor. *Computer* 34(11), 131–133.
5. Conboy, K. (2009) Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development. *Information Systems Research* (forthcoming).
6. Conboy, K. and B. Fitzgerald (2004). Toward a Conceptual Framework of Agile Methods. In: *Proceedings of Extreme Programming and Agile Methods – XP/Agile Universe 2004*. Springer, Berlin.
7. Derbier, G. (2003) Agile Development in the Old Economy. In: *Agile Development Conference*. IEEE Computer Society.
8. Dingsoyr, T., K.G. Hanssen, and T. Dybå (2006) Developing Software with Scrum in a Small Cross-Organizational Project. *EuroSPI Conference*, 5–15.
9. Drobka, J., D. Nofzt, and R. Raghu (2004). Piloting XP on Four Mission-Critical Projects. *IEEE Software* 21(6), 70–75.
10. Elssamadisy, A. and G. Schalliol (2002). Recognizing and Responding to “Bad Smells” in Extreme Programming. In: *Proceedings of the 24th International Conference on Software Engineering*. Association Computing Machinery, New York, 617–622.
11. Fredrick, C. (2003). Extreme Programming: Growing a Team Horizontally. In: *Extreme Programming and Agile Methods – XP/Agile Universe 2003*. Springer, Berlin, LNCS 2753, 9–17.
12. Lyytinen, K. and G. M. Rose (2006). Information System Development Agility as Organizational Learning. *European Journal of Information Systems* 15(2), 183–199.
13. Mann, C. and F. Maurer (2005). A Case Study on the Impact of Scrum on Overtime and Customer Satisfaction. *Agile 2005 Conference*, Denver.
14. Melnik, G. and F. Maurer (2004). Direct Verbal Communication as a Catalyst of Agile Knowledge Sharing. In: *Proceedings of the Agile Development Conference*. IEEE Computer Society, Los Alamitos, 21–31.
15. Miles, M. B. and A. M. Huberman (1994). *Qualitative Data Analysis: an Expanded Sourcebook*. Sage, Thousand Oaks, CA.
16. Poole, C. and J. Huisman (2001). Using Extreme Programming in a Maintenance Environment. *IEEE Software* 18(6), 42–50.
17. Rakitin, S. (2001). Manifesto Elicits Cynicism. *IEEE Computer* 34(12), 4.
18. Rising, L. and N. S. Janoff (2000) The Scrum Software Development Process for Small Teams. *IEEE Software* 17(4), 26–32.
19. Schatz, B. and I. Abdelshafi (2005) Primavera Gets Agile: A Successfull Transition to Agile Development. *IEEE Software* 22(3), 36–42.
20. Schwaber, K. (2003) *Agile Project Management with Scrum*. Microsoft Press, Washington.
21. Schwaber, K. and A. Beedle (2002) *Agile Software Development with SCRUM*. Prentice-Hall, Upper Saddle River, NJ.
22. Wang, X. and K. Conboy (2009) Understanding Agility in Software Development through A Complex Adaptive Systems Perspective, *ECIS 2009*.

The Influence of Short Project Timeframes on Web Development Practices: A Field Study

Michael Lang

Abstract A number of recent surveys of Web development have revealed that typical project timeframes are of the order of 3 months. This chapter reports the findings of a field study conducted in Ireland which set out to contribute towards a better understanding of the nature of high-speed Web development practices. Qualitative interview data was gathered from 14 interviewees, purposefully selected from a variety of different organisations and backgrounds. This data was then analysed using the Grounded Theory method, and ten core dimensions were revealed: (1) the role of collaborative groupware tools; (2) collective code ownership; (3) timeframe driven by business imperatives; (4) enablers of productivity; (5) quality “satisficing”; (6) requirements clarity; (7) process maturity; (8) collectively agreed project schedules; (9) closeness to client; and (10) working software over documentation.

Keywords Web development methodologies · Agile methods · Is project management

1 Introduction

Project timeframes can dictate the choice of a systems development method as well as the extent to which its various features may be used. A number of recent studies of Web-based systems development reveal that average delivery times are now about 3 months [1, 7], as further confirmed by the author’s own survey of Web development practices in Ireland [9]. A few years ago, this apparently hectic so-called “Web time” development context was alleged by Thomas [16] to give rise to “guerilla programming in a hostile environment using unproven tools, processes, and technology”. More recently, Baskerville and Pries-Heje [3, 4] found that short timeframes can lead to practices such as “coding your way out” and “negotiable quality”.

M. Lang (✉)

Business Information Systems Group, Cairnes School of Business & Economics, NUI Galway, Galway, Ireland

e-mail: michael.lang@nuigalway.ie

However, almost 70% of the respondents to the aforementioned survey reported having no or minor problems coping with the accelerated timescales of Web development, it being a “major” issue for just 4%. The motivation for this chapter was therefore to follow up on the survey findings with a qualitative field study which looked more closely at the nature of high-speed Web development practices.

2 Research Method

A field study consisting of semi-structured qualitative interviews with 14 Web designers/developers was conducted. A purposive, theoretical sampling approach was taken in selecting interviewees so as to seek out similarities and dissimilarities, looking at both typical and atypical cases [6, 12]. The profile of selected interviewees varied according to organisational size, organisational type (e.g. commercial/public sector or not-for-profit/private sector), nature of activities (e.g. Web design, digital multimedia design, “traditional” graphic design and/or “traditional” software development), application domains, location of end-users (in-house vs external clients) and professional background of interviewee (e.g. software development, graphic design, or other), as shown in Table 1.

With regard to application domain, we had observed in our previous survey [9] that certain industry sectors (e.g. Financial Services, Computer-Based Training) may be different from others as regards their use of development processes, methods and techniques. However, rather than selecting interviewees from a broad variety of different industrial sectors, it was decided to select a number of interviewees whose *clients* are from a broad variety of different sectors (i.e. Web development houses and design agencies) and to ask them how their development processes vary from one client to the next, if at all. A limitation of this approach is that there may be differences between, say, developing systems *for* a bank (outsourced development) as opposed to developing systems *within* a bank (in-house development), particularly where critical systems that require the application of specialised domain knowledge (e.g. advanced security) are not outsourced. That said, in selecting the interviewees for the field study, it was not feasible to contemplate all possible application domains and specialised considerations thereof, so the adopted strategy was seen as a good, if not optimal, way of eliciting data with regard to the influence of different application domains on process tailoring.

In most of the organisations visited, one personal interview was conducted with the team leader, typically convened during the mid-day break so as not to encroach upon busy work schedules. In one organisation two developers were separately interviewed, and in another the managing director brought five staff members into the meeting room. Where available, secondary data sources were also consulted. Data gathering continued until a point of reasonable “theoretical saturation” was reached. The data was analysed using a hybrid method, mainly based on the procedures of grounded theory [11, 14], but also informed by the principles laid down by Miles and Huberman [12].

Table 1 Profile of interviewees

Organisation	Industry (all private sector unless otherwise specified)	No. of employees	No. of developers	Interviewee job title	Interviewee background	Interviewee experience (years)
Bizweb	Web development	60	40	Managing Director + MIS Applications Architect + QA Manager	Business studies/Software development/Industrial engineering	Unknown
Clearscope DigiCrew	Web design agency	12	7	Senior Designer	Graphic design	7
	Web development	5	4	Internet Software Engineer	Computer games development	10
JobsPortal	On-line recruitment firm (in-house)	50	3	Web Project Manager	Software development	5
KL Design Martech	Graphic design	5	5	Managing Director	Graphic design	12
OEG	Web design agency	10	8	Creative Director	Graphic design	9
	University (public sector/in-house)	1,300	2	Chief Web Technologist	Software development	9
OEG	University (public sector/in-house)	1,300	2	Web Editor	Physics/Web development	10
Broadcorp	Broadcast media (public sector/in-house)	2,000	8	Web Project Manager	Industrial design	6
Redmoon	Visual communications	8	8	Managing Director	Film-making/Journalism	>10
Strata IBUS	Web development	7	5	Creative Director	Industrial design	10
	Web development	30	10	Managing Director	Software development	10
Webcorp W3M	Web development	25	20	Commercial Director	Business studies	10
	Web portal	2	1	Managing Director	Physics/Software development	15

3 Findings and Discussion

Our analysis of the interview data revealed ten core dimensions of high-speed Web development practices, as explained in the following sections.

3.1 Role of Groupware Tools as Enablers and Drivers of Collaborative Work

To support their project management process, Martech use an in-house workflow/job management tool where members of the team regularly update each other on each others' status. The use of similar groupware tools to support team collaboration was also mentioned by Webcorp, Bizweb, DigiCrew and IBUS, not just for project co-ordination but additionally for aspects such as code sharing, bug tracking and documentation of guidelines. A common pattern which emerged from interviews was that successful experimentation with open source or trial software often antecedes and expedites process definition, rather than the other way round. Otherwise put, the search for a simple, useful, shareable tool to address an ongoing problem that has attained chronic magnitude (e.g. requirements change control) can lead to efficient workable processes being built up around that tool. This is very much in line with Suchman's notion of "situated action" – upon which the conceptual framework of this dissertation is based – whereby "people use their circumstances to achieve intelligent action" [15].

3.2 Collective Code Ownership and Ease-of-Maintenance

At Bizweb, the organisation with the largest number of developers (40) of those visited, inefficiencies arising from the collective ownership of code – or more accurately, the lack thereof – has driven them to standardise their working methods and devise a mechanism whereby programmers can access and edit each others' code:

We've found in the past where somebody might end up wasting 5 or 6 days trying to work out and re-do somebody else's code. A lot of that was happening, it was ridiculous. Because people are going to get sick or take annual leave, so the continuity factor just didn't exist. It was you alone as regards your code, and that led to phone calls on holidays, which again leads to staff morale going down. So the [newly introduced] development procedures do work really well because it is trying to get everybody on the same track. And it forces you to examine your coding practices as well, and learn lessons about how things can be done better. (MIS Applications Architect, Bizweb)

This idea of collective code ownership, which notably is one of the tenets of Extreme Programming (XP) and other "agile" development methods, and the concomitant issue of ease-of-maintenance was also commented upon by the creative director at Martech:

Simplicity would be something that we would value a lot. Is the programming solution as simple as it can be? Are we inviting trouble down the line with this? Can another programmer pick this up and understand how it's written and why it's organised in this way? Equally, we would place importance even on the way designs are constructed. For example, a Photoshop document can be very complex with hundreds of layers, so it should be constructed in a way that makes it easy to re-use and modify, or for another designer to pick it up and key into. (Creative Director, Martech)

Another mechanism commonly used with the aim of achieving more cohesion and collective ownership within design teams is the use of regular morning briefings, a practice mentioned by quite a few of the interviewees which again is similar to the agile methods idea of daily stand-up meetings.

3.3 Timeframe Driven by Business Imperatives: Developer-Push and Client-Pull

It became apparent that, in many cases, the imperative to deliver systems quickly is as much if not more driven by the desire of Web design agencies to maximise throughput and revenue as by any sense of genuine urgency on behalf of clients. As such, most companies are working within the parameters of their own self-defined comfort zones. Notable exceptions are Broadcorp and JobsPortal, where pressing organisational deadlines are the norm and there is a very real and heightened sense of immediacy. In the majority of the other companies interviewed however, it seems to be more the case that clients tend to have fixed budget allocations, which in turn indirectly impose time constraints. As one developer explained:

The price is determined by how many days you spend on it, so if it costs, whatever, well then you work out how many days you can spend on that job, and that's all you can spend on it. Even if it's not fully done 100% correctly, it still has to go out . . . The developers and the designers will tell you that they want to finish the job perfectly, whereas the sales person and the project team will say 'you can't spend any more time on that job'.

This excerpt raises the problematic issue of negotiating compromises between time, quality and cost/resources; this potential trade-off and how it is managed are discussed later.

3.4 Enablers of Rapid Development and Enhanced Productivity

To state an obvious point, short delivery cycles in Web-based systems design have become the norm because they are *possible*. Otherwise put, the factors which enable rapid development also serve to raise expectations and therefore drive demand, be that from clients or from project managers. These enabling factors stem from a variety of sources. First, Web-based systems can be rapidly deployed because the Web is an immediate delivery medium that, unlike traditional IS and off-the-shelf software applications, is not impeded by production, distribution and installation delays. Second, there have been dramatic gains in recent years in developer productivity.

This is facilitated in the first instance by the availability of high-speed rapid application development tools for Web development, e.g. ColdFusion. Third, it has become common practice to make extensive re-use of libraries of pre-fabricated components and applets; templates and wizards for automatic code generation; plug-and-play interfaces for database connectivity; and customisation of ready-made open source solutions. This has been refined to a point where most development time is now invested into the ongoing evolution of an “out-of-the-box” “productised” solution, such as advanced content management functionality. Code production has moved from crude cut-and-paste re-use to instant automatic generation, meaning that most of the standard back-end functionality required for any given project can be up and running within a day or two. As an example of the scale of productivity improvements which have been achieved by the use of rapid development tools, automatic code generation and systematic re-use, DigiCrew can now do in a week what would have taken 2 months to do just 3 years ago.

Fourth, the idea of “picking the right tool for the job”, meaning the one that can get it done as best and as efficiently as possible, was a recurrent pattern:

If the tool is good at a certain thing you will normally rely on the tool an awful lot more for that particular thing, and you will normally re-arrange the way you do things so that you will use that tool fully before moving onto another aspect. (Chief Web Technologist, OEG)

Fifth, just as software components can be re-used, the re-use of graphic design elements also speeds up development. Previous research has suggested that graphic designers, being of a “creative” disposition, are not inclined to re-use previous work [10]. The findings of this field study indicate the contrary. This discrepancy can be explained by clarifying what the concept of “re-use” means to a graphic designer. Whereas in software development, a piece of code might be entirely re-used as-is from a previous project, in graphic design a previously used component would constitute a useful “starting point”, but that component would always be uniquely re-worked to some extent. From their background tuition in art, graphic designers are trained to seek and synthesise elements and styles from various sources. As such, the re-use of concepts is actually a normal part of their work:

I’ve built up a database of sites that I like. When I come across something, I’ll take a screen grab and store it as a JPEG, so when I’m looking for an idea for a new site, I can go through maybe 200 images that are from previous sites, something that might have the same colour scheme or be the same sector or whatever, and I could use that then as a starting point for a new design. (Managing Director, KL Design)

Lastly, a factor which improves developer productivity is know-how and expertise. Baskerville and Pries-Heje [3] identified “dependence on good people” as one of the elements of their “e-Methodology” for rapid Web-based systems development. However, as has been recently debated with regard to both open source software development and agile methods, practices which are reliant upon such a rare commodity as naturally talented programmers are not sustainable. The companies interviewed in this field study were mostly industry leaders who have received numerous professional awards. Successful companies arguably have the advantage of being able to attract better staff, but that success in the first instance is predicated

upon the quality of existing staff. While these award-winning companies shared the characteristic that they were all led by highly motivated and talented individuals, they mostly also share a common concern with the management of design knowledge. Important types of knowledge mentioned were: application domain knowledge, knowledge about development tools/environments and technical standards, knowledge about design methods and techniques, knowledge of core design principles and a repertoire of time-efficient work-arounds. Most award-winning companies have mechanisms in place to facilitate and encourage the management of such knowledge (e.g. intranet bulletin boards, “Wiki’s” and “blogs”), with rewards and bonuses accruing to employees who use slack time to acquire and exchange useful knowledge. A number of them also schedule regular time slots for research activity, setting aside normal development work.

3.5 Impact of Time Pressure on Quality: Extensive Re-use, “Pragmatic Satisficing”

The planning/requirements definition phase is the most time-intensive part of the Web-based systems design process. These aspects are of critical importance, for as Brooks [5] puts it, “no other part of the work so cripples the resulting system if done wrong”. For large-scale systems with many different classes of users, it is preferable to perform user-centred design and to conduct a thorough needs analysis if time permits. However, in some cases, it does not. The time–cost–quality trade-off is a well-known phenomenon in software development. In their study of high-speed software development, Baskerville et al. [2] found that “when time drives development, product quality along with performance and cost, assumes second priority”. They labelled this concept “negotiable quality”, by which is meant that “customers and users seem to expect low quality” because of time pressure [3].

In spite of project delivery cycles being of the order of 6 weeks, very little evidence of “negotiated quality” was found in this field study. One possible explanation is that, in the wake of the post-Y2K “dot.com” industry shake-up, the marketplace has become more competitive and users are much less tolerant of unprofessional standards of work. For such vital aspects of system quality as response time, reliability, ease-of-use, visual attractiveness and security, excellence is a commercial imperative. Given the demand for high-quality, low-cost productions in short timeframes, firms within the Web design industry have adapted their practices to extensively avail of re-usable pre-tested components, making it possible to rapidly develop reliable and robust systems:

Our content management system is now in phase 1.6. There’s a new version every few months, and it gets fully tested before the new phase goes live. Let’s say you come to me and say ‘I want a Web site’. We have a function on our system where you just press a button, enter some basic parameters into a form, and fill in the HTML templates for the header and the footer. So instantly we can launch a fully proven Web site because the system has already been built. So a lot of the need for testing has been taken out because of the productisation. (Managing Director, IBUS)

In situations where deadlines are pressing and available resources are tightly constrained, what can happen is that a sub-optimal, but nevertheless acceptably good, solution is delivered. This practice, as exemplified by the following excerpt, might be called “pragmatic satisficing”, a form of “negotiated quality”:

What often happens is that the producer of the show has 2 or 3 weeks to get it done, they're 'busy, busy, busy', and then they go 'Hey, we should have a Web site'. So then at least you get everything all in one lump, but you might only have a few days notice. All we can really do is put out a formalised, *set* kind of a Web site, based on one of our standard templates. (Web Project Manager, Broadcorp)

It is the combination of acute time and resource constraints that leads to this practice of pragmatic satisficing. At Broadcorp, which is in the media industry, these pressures are especially pronounced; as the Web project manager explained, their concept of “time” is measured in hours, not days, while resources to hand are scarce and fixed. At OEG, resources are similarly limited, but there is not the same commercial imperative to deliver systems quickly so quality is not compromised. At JobsPortal, projects with high urgency are handled by bringing in hired contractors, and/or by re-negotiating the relative priorities of backlogged projects with management (negotiated schedules). Elsewhere, the strategies typically employed by commercial Web design agencies to manage this scenario of acute time and resource constraints are either to haul in the client's expectations (negotiated scope) or to outsource some work, and before all of this there is also a widespread practice of factoring buffer time into project estimates. In all of these approaches, quality is the paramount concern and it is not subverted, as Baskerville et al. [2] suggest, by time and cost considerations. Even in the worst-case scenario, referred to herein as the tactic of “pragmatic satisficing”, a tried-and-tested solution is delivered, albeit it is neither “award winning” (Strata) nor “progressing the boundaries of design” (Broadcorp).

3.6 Requirements Clarity: Need to “Freeze” and Sign Off

The clarity and stability of requirements is an age-old issue in systems development, so it was not surprising that our earlier survey had found the most acute problem in Web-based systems design to be the control of scope/feature creep. The other principal challenge revealed by the survey was the preparation of time and cost estimates [9]. Of course, time/cost overruns and scope creep are intrinsically linked. A major cause of scope creep is that projects often kick off with a very vague idea of the requirements. As one interviewee explained,

When you go into a pitch for a job you'll say 'Yeah, we can turn this around in 6 weeks', but at that stage you don't know what their specific requirements are. So that timeline will be altered after we find out what they want. (Creative Director, Strata)

Baskerville and Pries-Heje [3] had previously noted this, making the point that “an inability to pre-define system requirements is the central, defining constraint of Internet time development”. The pattern which emerged from interviews is that

when clients make initial contact with designers, they typically have little more in mind than a loose set of aspirations. In the initial meeting, these are usually documented in a one- or two-page brief, which also captures such essentials as: project budget, timeframe, main competitors, target audience and project goals. A detailed requirements specification is then produced by way of negotiation over the course of a number of meetings.

In general, the requirements specification document seems to be predominantly the vision of the designers, wherein they describe what they can do for the client, taking resource allocations into consideration. In all the commercial Web design agencies that were interviewed, the dominant constraint is usually the client's budget. Clients often have naïve expectations at the outset so the sales team, after consultation with the project team, must come to an arrangement as to what can be delivered, for what price, within what timeframe. Though most of the functional requirements are typically standard and can therefore be readily described and costed "à la carte", the bespoke elements take time to specify, as does a considered analysis of the fine details of the overall package including the "non-functional" requirements (usability, accessibility, security, performance levels, etc.). As initially revealed by the survey and later substantiated by follow-up interviews, it is common practice to produce and "freeze" a detailed requirements specification before commencing full-scale production.

These requirements specifications are essentially pseudo-legal bargaining chips that are used to control creep, cost and scheduling, but they also serve a defensive purpose whereby project managers can insulate themselves from political fallout by insisting upon a clear signed-off brief:

I'm responsible for delivering projects on time. If anything goes wrong, I'm answerable for my team. My team has to deliver to me, and I am answerable to the top people . . . If there is a communication gap between the users and the developers, the project doesn't go well! That's why we try to get it signed off as much as possible. If it is not signed off, we could be in trouble. (Web Project Manager, JobsPortal)

3.7 Streamlined Processes and Procedures to Support a Sustainable Pace

Jayaratra et al. [8] make the point that "methodologies are time ordering mechanisms". Where project cycles are customarily tight, it seems reasonable to expect that time-efficient working procedures would be in place. In the survey [9], it was found that processes tend to be more formalised and explicit in Web development companies than in traditional IT/software development companies. A possible explanation for this which emerged in the interviews is the sales-driven high-speed nature of work practices in Web design agencies:

You have to streamline how you do things. You have to build processes, put them in place, and just *follow* them . . . So when a Web design project comes in, you know exactly what to do, you take it, and you go bang-bang-bang-bang. (Web Editor, OEG)

Consistent with the results of Baskerville and Pries-Heje's [3] study, it was found that much Web design/development work is done in parallel, similar to the notion of "concurrent engineering" in manufacturing, thereby speeding up development times.

In high-speed development environments, an important issue is how to support a sustainable pace whereby the project team consistently manages to deliver short-cycle projects on time without their stamina being diminished. Here again, streamlined work procedures can be beneficial, as exemplified by the experiences at Bizweb:

When you're doing makeshift things you end up reinventing the wheel, which isn't cost effective or productive. And then that leads to you overworking your staff, morale is low, and you can't motivate them. If you ask some people how the company has improved in the past 2 years, they'll say 'Whoa, in the old days we had to . . .', it all ties in with standardising things and putting procedures in place. (QA Manager, Bizweb)

3.8 Project Management: Collectively Determined Schedules, Cohesive Teams

Where project timeframes are short, it is important that time estimates are accurate because small overruns, in relative terms, are more significant. In the survey, it was found that, despite the intrinsic difficulties in preparing time and cost estimates, Web project managers are faring quite well.

In their study of Web design practices, Rodriguez-Garcia and Harrison [13] found that project management estimates are most commonly formed by analogy and judgement, and also that most organisations collect time/effort metrics as timesheets for billing. The findings of this field study reveal a similar picture. Nearly all of the commercial Web design agencies/development houses visited spoke of the use of job management systems wherein *collectively* agreed estimates are recorded and change requests are logged. By asking developers to set their own schedules, those schedules are more likely to be reasonable, therefore facilitating a sustainable pace. This practice contributes to enhanced staff morale because it reduces the need for overtime and also because the development team are empowered to determine and take personal responsibility for their workloads.

The opposite effect was also found to exist. As explained by two different interviewees, each speaking of practices in former places of employment, project time estimates which are dictatorially imposed rather than democratically negotiated can lead to resentment and coercion, and ultimately to staff turnover:

Of all the issues, – people are even leaving their jobs because of it, – the most serious one is down to project management. Let's say, the job comes in, the project manager talks with the client, the sales guy signs the deal. That's the timeline. It's agreed with the Web development team, and everyone's happy with that. An issue arises with the customer where the project manager changes the dates according to the plan without even discussing with the Web team, and that causes mayhem. Because it's a very tight process, even if you move things out by a day or two that will affect other jobs, and it just becomes a mess.

The sales people and the project managers can cause *hell* for developers by over-promising, by not understanding what's involved and not consulting with the developers on the project timelines . . . And unfortunately what seems to happen is that programmers roll over, they work all hours to meet these deadlines, and that's not noticed by management, but when they complain that they're overworked, the management typically just say 'Well, you've done it before, you mustn't really mean it!'.

Whereas Web design agencies prepare detailed timelines and breakdowns – such as at Webcorp where the commercial manager explained that a typical project plan if printed out would cover one whole side wall of his office – a somewhat different picture emerged for the in-house development teams. At Broadcorp and JobsPortal, “elegant” project plans are not drawn up on such a grand scale simply because relative priorities are driven by the organisation's business imperatives which can change dramatically from day to day, so they must be very flexible and responsive. Web design agencies typically operate within comfort zones, making allowance for a certain amount of slack. As laid out in project task breakdowns and agreed work schedules, individual team members can focus their attention on specific projects for dedicated blocks of time. In contrast, in-house development teams usually find themselves facing multiple urgent deadlines with little room to manoeuvre. Interestingly, the Web project manager at Broadcorp used the metaphor of a “flight controller” to describe how he copes with this challenge:

It's coming to the stage where it's just like landing planes. There's 10 or 15 projects flying around up there, so you just pick the one that needs to be done. (Web Project Manager, Broadcorp)

3.9 Closeness of Relationship with Client Project Team

Rather like the agile methods concept of an “on-site customer”, it was found that a prerequisite for rapid Web-based systems development is a close relationship with the client, as well as the unity and commitment of the client organisation's project team. In the absence of any of these factors, communication becomes protracted and jumbled, inevitably causing project delivery times to slip:

If a client sits on something for a couple of days, the project will be delayed. Schedules are like concertinas all the time. We can tell a client that we'll have this done for you in 6 weeks, but that often gets pushed out because we're waiting for something to come back . . . There's one project we're working on, it's been stalled now for a year because on the client side they've no real project manager. They have about 15 people looking after the various elements of the Web site so no-one knows who's doing what and they've all got other more important things to do. (Creative Director, Strata)

This pattern of procrastination and aimlessness is exacerbated where the dreaded phenomenon of “design by committee” presents itself, as one interviewee explained:

The minute you're in-house, you're prone to more politics. In my last job, I had one boss who made all the decisions when it came to anything. That doesn't happen in here, it's all committee-based. Sometimes it's extremely frustrating, – when you're pushing forward

with a project, you might have to stop for three weeks or longer just to wait for someone to take a look. It can really disrupt and is probably slowing things down ten-fold.

3.10 Working Software over Documentation

When you engage with clients they want to see something, they want to get programming started. But we'll say 'now hold on, the most important job is the definition document, you agree the plans with a builder who builds a house, so we're not even going near the bricks until this is done'. (Managing Director, IBUS)

Though in most cases, as in the above excerpt, interviewees are firmly of the opinion that jumping straight to coding without a robust design specification is ill-advised, there is also a widely held view that the production of documentation is simply a means to an end and that beyond a certain point of "good enough" it becomes a resource-sapping, non-value-adding, unnecessary activity. Thus, the value of light, essential documentation is accepted, but given the imperative to turn projects around quickly, prototypes and working software is developed as early as possible, and refactored and evolved as required depending on change requests:

Not necessarily straight away, but I think as early in the process as possible you should start coding. If you have a good idea of what you're doing, say you've got 70% or 80% of the requirements tied down, I'd be inclined to move on. I suppose that's more to do with my background as a software engineer, I would be itching to get into the actual implementation of it as early as possible, I'm not a huge fan of too much paperwork although I think it is important to capture the gist of the functional specification. (Managing Director, W3M)

4 Conclusions

Consistent with the previous work of Baskerville and Pries-Heje [3, 4], this study found, as one would expect, that time pressure is the central determinant of design practices. However, there are discrepancies between this research and that of Baskerville and Pries-Heje, most notably with their finding that developers may resort to the practices of "coding your way out" and "negotiated quality" because of the pressures of high-speed development environments. Whereas in Baskerville and Pries-Heje's study such practices were endemic, in this research hardly any such incidents were discovered. This can be explained in a number of ways. First, the interviewed companies were mostly award winners, a likely indicator that they make special efforts to strive for excellence and quality. Second, the marketplace has become more competitive in recent years and users are much less tolerant of unprofessional standards of work, meaning that expectation levels have risen. Third, the use of pre-fabricated "productised" solutions that are already fully tested means that robust systems can be rapidly delivered without compromising cost or quality. Even in the worst-case scenario for a development team, where

they face the dreaded “backs-to-the-wall” combination of acute time and resource constraints, a tactic herein coined as “pragmatic satisficing” is engaged, meaning that a tried-and-tested solution is re-used, albeit it may not be the best possible outcome.

Given the high-speed nature of Web-based systems development, the emphasis of development practices is very much on agility, speed, efficiency and productivity. Streamlined processes are necessary in order to maximise throughput, and also to sustain a continual pace by eradicating the need for ongoing overtime (which has fatiguing and demoralising effects). Interestingly, many of the Web developers interviewed have evolved practices that are markedly similar to those of the “agile” methods family, such as collective code ownership; an emphasis on simplicity; the use of regular informal team briefings; insistence on a close working relationship with the client; the pursuit of continuous process improvement through reflective evaluation; and a general emphasis on people, communication and working software over processes, documentation and adherence to a plan.

References

1. Barry, C. and Lang, M. (2003) A Comparison of “Traditional” and Multimedia Information Systems Development Practices. *Information and Software Technology*. 45(4), 217–227.
2. Baskerville, R., Levine, L., Pries-Heje, J., Ramesh, B., and Slaughter, S. (2001) How Internet Software Companies Negotiate Quality. *IEEE Computer*. 34(5), 51–57.
3. Baskerville, R. and Pries-Heje, J. (2001) Racing the E-Bomb: How the Internet Is Redefining Information Systems Development Methodology. In: Russo, N. L. et al. (eds), *Realignment Research and Practice in Information Systems Development: The Social and Organizational Perspective*. IFIP WG8.2 Conference, Boise, Idaho, USA, 27–29 July 2001, pp. 49–68. Boston: Kluwer Academic Publishers.
4. Baskerville, R. and Pries-Heje, J. (2004) Short Cycle Time Systems Development. *Information Systems Journal*. 14(3), 237–264.
5. Brooks, F. P. (1987) No Silver Bullet/Essence and Accidents of Software Engineering. *IEEE Computer*. 20(4), 10–18.
6. Glaser, B. G. and Strauss, A. L. (1967) *The Discovery of Grounded Theory: Strategies for Qualitative Research*. New York: Aldine de Gruyter.
7. Glass, R. L. (2001) Who’s Right in the Web Development Debate? *Cutter IT Journal*. 14(7), 6–10.
8. Jayaratna, N., Holt, P. and Wood-Harper, T. (1999) Criteria for Methodology Choice in Information Systems Development. *Journal of Contemporary Issues in Business and Government*. 5(2), 30–34.
9. Lang, M. and Fitzgerald, B. (2005) Hypermedia Systems Development Practice: A Survey. *IEEE Software*. 20(2), 68–75.
10. Linden, T. and Cybulski, J. (2003) Capturing Best Practices in Web Development. In: Isaías, P. and Karmakar, N. (eds), *IADIS International WWW/Internet 2003 Conference, Algarve, Portugal, November 5–8, Vol. 1*, pp. 427–434. Lisbon, Portugal: IADIS Press. ISBN 972-98947-1-X.
11. Locke, K. (2001) *Grounded Theory in Management Research*. London: Sage.
12. Miles, M. B. and Huberman, A. M. (1994) *Qualitative Data Analysis: An Expanded Sourcebook*, 2nd Edition. Thousand Oaks, CA: Sage.

13. Rodriguez-Garcia, D. and Harrison, R. (2000) Practitioners Views on Web Development: An Industrial Survey by Semi-Structured Interviews. In: 13th International Conference on Software and Systems Engineering and Their Applications (ICSSEA 2000), Paris, France, December 5–8. CNAM. Paris, France.
14. Strauss, A. and Corbin, J. (1998) Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory, 2nd Edition. Thousand Oaks, CA: Sage. ISBN 0-8039-5939-7.
15. Suchman, L. A. (1987) Plans and Situated Actions: The Problem of Human-Machine Communication. Cambridge: Cambridge University Press.
16. Thomas, D. (1998) Web Time Software Development. Software Development Magazine. 6(10), 78–80.

On Weights Determination in Ideal Point Multiattribute Decision-Making Model

Xin-chang Wang and Xin-ying Xiao

Abstract TOPSIS is a commonly used method in multiattribute decision making. With the weight standardization matrix, it ranks the projects through/by calculating the distances of each project to the positive ideal point and to the negative ideal point. In such a method, the key problem is how to decide the weight of each attribute. First, the chapter analyzes the deficiency in former studies about attribute weights determination; and second, the author proposes a weight determination method based on the principal components. This method decides weights of attributes according to their contribution in sample data. So, the influence of subjective factors can be reduced, the deviation between projects choice can be avoided, and the real importance of any attribute can be reflected objectively. This method entrusts great weights to indexes which synthesize much sample information and small weights to indexes which synthesize little sample information. It conforms to the basic meaning of indexes weights.

Keywords Factor analysis · Ideal point · Weight determination

1 TOPSIS

TOPSIS, “technique for order by similarity to ideal solution” [1], is a common method in multiobjectives finite projects decision-making analysis. It has characteristics of easy computation, reasonable result, and wide application. In this method, the projects are filtrated with the aid of “the positive ideal point” and “the negative ideal point.” The positive ideal point is a supposition best project, whose various attribute values all achieve best in each candidate project, and the negative ideal point is another supposition worst project, whose various attribute values all achieve

X.-c. Wang (✉)

Information School of Jiangxi University of Finance & Economics; Mathematics & Application
Mathematics Department of Jinggangshan University, Jiangxi, China
e-mail: wangxinchang11@163.com

worst. By calculating the weight distances between each project to the best project and to the worst project, the approach extent of each project to the best project can be obtained, and then the order of all projects are gained.

In a multiattribute problem, suppose an alternative subject set is $A = [a_1, a_2, \dots, a_n]$ and an attribute set is $X = [X_1, X_2, \dots, X_m]$, so, an attribute value of a project a_i is $\{x_{i1}, x_{i2}, \dots, x_{im}\}$, which is a point of m -dimension space, and the sample decision-making matrix is $X = [x_{ij}]_{n \times m}$. Then the steps of TOPSIS are described as follows:

- (1) Standardize the sample data using vector standardization methods. Set

$$y_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^n x_{ij}^2}}$$

We get the standardization data matrix $Y = [y_{ij}]_{n \times m}$ ($i = 1, 2, \dots, n; j = 1, 2, \dots, m$)

- (2) Form weight standardization matrix Z .

Given the attribute weight $W = (w_1, w_2, \dots, w_m)$, and set $z_{ij} = w_j y_{ij}$, w_j is the weight of the j th attribute, ($i = 1, 2, \dots, n; j = 1, 2, \dots, m$), we can get the weight standardization matrix $Z = [z_{ij}]_{n \times m}$.

- (3) Decide the positive point and the negative point.

In the weight standardization matrix Z , choose the best attribute value to structure the positive ideal point $Z^+ = (z_1^+, z_2^+, \dots, z_n^+)$, and choose the worse attribute value to structure the negative ideal point $Z^- = (z_1^-, z_2^-, \dots, z_m^-)$.

- (4) Calculate the distance of each project to the positive ideal point $d_i^+ = \sqrt{\sum_{j=1}^m (z_{ij} - z_j^+)^2}$, and calculate the distance to the negative ideal point $d_i^- = \sqrt{\sum_{j=1}^m (z_{ij} - z_j^-)^2}$, $i = 1, 2, \dots, n, j = 1, 2, \dots, m$.
- (5) Calculate the extent of each project approach to the positive ideal point $c_i = d_i^- / (d_i^+ + d_i^-)$, $i = 1, 2, \dots, n$, and determine the order of all projects.

We can see, in the above operation steps, the determination of the weights of attributes is very crucial. If the weights of attributes are determined unreasonably, it will bring deviation in the choice of projects.

2 The Weights Determination Problem of TOPSIS

In TOPSIS, there are usually two methods in the determination of attribute weights. One is determined by decision makers with their experiences and the other is determined by professional experts by using AHP method. Whether it is obtained by decision maker's experiences or by experts, there is some inevitable subjectivity and randomness in these two methods, and it will bring deviations and faults in choosing decision projects.

In order to reduce the subjectivity and the randomness in attribute weight determination, Xu Zeshui [2, 3] of Southeast University proposed TOPSIS based on the optimized model. His main ideas are described as follows:

(1) In the standardization matrix, set the positive ideal point as $Z^+ = [1, 1, \dots, 1]$ and the negative ideal point as $Z^- = [0, 0, \dots, 0]$. Suppose the attribute weight vector will be determined as $W = (w_1, w_2, \dots, w_m)$.

(2) The more the project approaches the positive ideal point, the better the project is. We can set the weight deviation between the project a_i to the positive ideal point as

$$e_i^+(w) = \sum_{j=1}^m |y_{ij} - 1| w_j = \sum_{j=1}^m (1 - y_{ij}) w_j$$

Here, $Y = [y_{ij}]_{n \times m}$, $i = 1, 2, \dots, n$; $j = 1, 2, \dots, m$ is a standardization data matrix.

(3) For a given weight vector $W = (w_1, w_2, \dots, w_m)$, the smaller $e_i^+(w)$ is, the better the project a_i is, so we can set up multiobjectives decision-making model as:

$$\begin{cases} \min e^+(w) = (e_1^+(w), e_2^+(w), \dots, e_{n1}^+(w)), \\ \text{s.t. } w_j \geq 0, \quad \sum_{j=1}^m w_j = 1 \end{cases}$$

As each project competes impartially, and furthermore as there is no preference in all projects, we can summarize the above model as the following one-objective model:

$$\begin{cases} \min e^+(w) = \sum_{i=1}^n e_i^+(w), \\ \text{s.t. } w_j \geq 0, \quad \sum_{j=1}^m w_j = 1 \end{cases} \quad (1)$$

By solving this programming problem, we can get the attribute weight vector $W = (w_1, w_2, \dots, w_m)$.

(4) Put the weight vector $W = (w_1, w_2, \dots, w_m)$ into $e_i^+(w)$.

In the above method, any influences of human factors are eliminated in weight attributes determination, but it brings other questions: one problem is the existence of optimal solution. Even if the one-objective programming has optimum solutions, and the optimal weight vector $W = (w_1, w_2, \dots, w_m)$ is obtained, the attribute weight vector makes the weight deviation of each project to the positive ideal point smallest, but it is obtained not by comparing the importance of attributes, and such attribute weights will vary with sample data.

For example, there is a problem about evaluating enterprises innovation ability in one district. Suppose the attributes are: the rate of R&D inputs to sales income (x_1), the rate of technology staff to employees (x_2), the level of technology equipment (x_3), the rate of new products sales income to product sales income (x_4), the rate of

Table 1 Attribute sample values for five enterprises

Attribute	Enterprise 1	Enterprise 2	Enterprise 3	Enterprise 4	Enterprise 5	Enterprise 6	Enterprise 7
X1(%)	23.1	15.6	38.4	24.1	56.1	X1(%)	23.1
X2(%)	5.3	2.3	8.9	12.5	15.2	X2(%)	5.3
X3(%)	32.5	35.6	15.6	25.6	45.2	X3(%)	32.5
X4(%)	52.3	21.6	45.6	28.9	30.8	X4(%)	52.3
X5(%)	45.3	20.6	45.3	27.6	34.8	X5(%)	45.3
X6(unit)	45	12.	16	48	23	X6(unit)	45
X7(%)	10.2	5.6	2.3	0.9	1.2	X7(%)	10.2

Table 2 Attribute sample values for seven enterprises

Attribute	Enterprise 1	Enterprise 2	Enterprise 3	Enterprise 4	Enterprise 5	Enterprise 6	Enterprise 7
X1(%)	42.3	32.1	24.5	36.8	49.3	50.1	48.2
X2(%)	10.2	15.3	16.2	4.5	14.2	5.6	24.3
X3(%)	32.1	15.6	24.3	36.2	14.5	45.1	16.8
X4(%)	45.6	32.4	48.1	24.1	25.9	24.8	19.2
X5(%)	40.2	23.6	45.2	12.6	10.3	25.9	16.3
X6(unit)	54	42	13	15	46	48	15
X7(%)	0.8	1.9	8.9	3.6	0.9	1.8	2.5

new products sales profit to product sales profit (x_5), the number of invents or patents (x_6), and the rate of technology income to sales income (x_7). The sample decision-making matrix for evaluating five and seven enterprises are as follows (shown in Tables 1 and 2):

By standardizing the above two decision matrixes, and then putting them into model (1), we can get the attribute optimum weights, respectively, as:

$$W = (w_1, w_2, \dots, w_7) = (0.1435, 0.1408, 0.1482, 0.1484, 0.1495, 0.1412, 0.1285),$$

$$W = (w_1, w_2, \dots, w_7) = (0.1494, 0.1427, 0.1449, 0.1470, 0.1424, 0.1423, 0.1313)$$

We can see, for evaluating the enterprise innovation ability, the attributes are the same in the two problems, but the weights are not the same, and this is not reasonable.

3 Attribute Determination Based on Principal Component

In order to reduce the subjectivity and the randomness in the attribute weight determination, to avoid the disadvantage of weights varying with sample data, and

to make weights truly reflect the importance of attributes, author brings up the following weights determination methods based on principal components.

The idea of principal components method is merging some original relative indexes $X_j (j = 1, 2, \dots, m)$ into new nonrelative comprehensive indexes F_k (namely, principal components) ($k = 1, 2, \dots, p$), using F_k index system to replace X_j index system, and using the variance of each principal components $\text{Var}(F_k)$ to express information it integrates original sample information. The larger $\text{Var}(F_k)$ is, the more original sample information the k th principal component contains, and it means F_k is important in all principal components. Therefore, we can use the rate of variance of the k th principal component $\text{Var}(F_k) / \sum_{k=1}^p \text{Var}(F_k)$ to express the weight of the k th principal component. The author describes attribute weights determination method based on principal components as follows.

3.1 Standardize the Sample Decision-Make Matrix

Suppose the sample decision-making matrix is $X = [x_{ij}]_{n \times m}$.

To eliminate the influence of dimension among different indexes, the influence of positive and negative direction, standardize the sample data according to the following formula:

$$y_{ij} = \frac{x_{ij} - \bar{x}_j}{\sigma_j}, \quad \bar{x}_j = \frac{1}{n} \sum_{i=1}^n x_{ij}, \quad \sigma_j = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_{ij} - \bar{x}_j)^2}$$

We can get standardization matrix $Y = (y_{ij})$.

3.2 Calculate Eigenvalue and Eigenvector of the Correlation Coefficient Matrix

Set a linear combination of vectors $Y_1, Y_2, \dots, Y_m$, the column vectors of standardization matrix $Y = (y_{ij})$.

$$\begin{cases} F_1 = a_{11}Y_1 + a_{21}Y_2 + \dots + a_{m1}Y_m \\ F_2 = a_{12}Y_1 + a_{22}Y_2 + \dots + a_{m2}Y_m \\ F_m = a_{1m}Y_1 + a_{2m}Y_2 + \dots + a_{mm}Y_m \end{cases}, \quad \text{where } Y_j = \begin{bmatrix} y_{1j} \\ y_{2j} \\ \vdots \\ y_{nj} \end{bmatrix}$$

$F_1, F_2, \dots, F_m$ are defined m principal components. We hope that in these principal components, the more front the component is, the more information it contains. The variance of each principal component shows how much it integrates original

information, so the principal components $F_1, F_2, \dots, F_m$ must satisfy the following conditions:

(1) F_i is not related to $F_j (i \neq j, i, j = 1, 2, \dots, m)$.

(2) F_1 is the largest variance component in all linear combinations of $X_1, X_2, \dots, X_m$; F_2 is the largest variance component in all linear combinations of $X_1, X_2, \dots, X_m$ which are not related to F_1 ; $\dots$; F_m is the largest variance component in all linear combinations $X_1, X_2, \dots, X_m$ which are not related to $F_1, F_2, \dots, F_{m-1}$.

It can be testified that the coefficient vector $(a_{1i}, a_{2i}, \dots, a_{mi}), i = 1, 2, \dots, m$ of the principal components $F_1, F_2, \dots, F_m$ which satisfies the above conditions is exactly the eigenvectors of matrix $\sum$, the covariance matrix Y . When the covariance matrix $\sum$ is unknown, we can use its sample estimation value S (sample covariance matrix) to replace it.

$$S = (s_{ij}) \text{ and } S = (s_{ij}) \quad \text{and} \quad s_{ij} = \frac{1}{n} \sum_{k=1}^n (x_{ki} - \bar{x}_i)(x_{kj} - \bar{x}_j)$$

The correlation coefficient matrix is $R = (r_{ij})$ and $r_{ij} = s_{ij} / \sqrt{s_{ii}} \sqrt{s_{jj}}$.

As $Y_1, Y_2, \dots, Y_m$ have been standardized, we can get $S = R = (1/n) Y^T Y$.

Because matrix $Y^T Y$ and $(1/n) Y^T Y$ only differ with a coefficient, it is obvious that the eigenvalues of matrix $Y^T Y$ is n times of those of matrix $(1/n) Y^T Y$, and their eigenvectors will not vary, and therefore will not affect principal components determination; we set $R = Y^T Y$ simply.

By solving the characteristic equation $|\lambda E - R| = 0$, we obtain m eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_m$ of correlation coefficient matrix R . Ranking the eigenvalues as $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m \geq 0$, and getting eigenvectors $(a_{1i}, a_{2i}, \dots, a_{mi}), i = 1, 2, \dots, m$ correspondence to each eigenvalue λ_i from equation $(\lambda_i E - R)X = 0$, we defined $\lambda_1 / \sum_{i=1}^m \lambda_i$ as contribution rate of the first principal component, which is the variance percent ratio of the first principal component. The larger this ratio is, the more original information of attributes $X_1, X_2, \dots, X_m$ the first principal component contains. The accumulated contribution rate of first two principal components is defined as $(\lambda_1 + \lambda_2) / \sum_{i=1}^m \lambda_i$, and the accumulated contribution rate of first k principal component is defined as $\sum_{i=1}^k \lambda_i / \sum_{i=1}^m \lambda_i$. If the accumulated contribution rate of first p principal component reaches more than 85%, it shows that the first $p (p \leq m)$ principal components basically integrate original sample information. In this way, not only is the number of attribute variables reduced but the decision-making problem is also analyzed easily.

3.3 Obtain the Weights of p Principal Components

The p principal components $F_1, F_2, \dots, F_p$ which their accumulated contribution rate reaches more than 85% must be defined as attributes in TOPSIS, and the matrix $Z = [F_1 \ F_2 \ \dots \ F_p]_{n \times p}$ must be defined as the standardization matrix, so the weight vector of $F_1, F_2, \dots, F_p$ is

$$W = (w_1, w_2, \dots, w_p) = \left(\lambda_1 / \sum_{i=1}^m \lambda_i, \lambda_2 / \sum_{i=1}^m \lambda_i, \dots, \lambda_p / \sum_{i=1}^m \lambda_i \right),$$

that is to say, the weight of each principal component is the variance contribution rate of each principal component.

4 Conclusions

In this chapter, in one decision-make problem, by choosing reasonably linear combination, we can integrate $X_1, X_2, \dots, X_m$ into p new nonrelative principal components, and these p new nonrelative principal components almost contain the original attribute information. By calculating variance contribution rate, the weights of p principal components are obtained, and so the weights of attributes in TOPSIS are obtained. This method of weight determination is entirely based on sample data information. So, the influence of subjective factors can be reduced, the deviation between projects choice can be avoided, and the real importance of any attribute can be reflected objectively. This method entrusts great weights to indexes which synthesize much sample information, and small weights to indexes which synthesize little sample information. It conforms that the attribute weight must be coincident to the importance of attribute.

References

1. Yue, C. (2004) Decision Making Theories and Methods, Beijing: Science Publishing Company, pp. 192–244.
2. Xu, Z. (2004) Uncertain Multiple Attribute Decision Making: Methods and Its Application, Beijing: Qinghua University Publishing Company, pp. 76–102.
3. Qian, G. and Xu, Z. (2003) The Three Optimum Models Based on TOPSIS in Uncertain Multiple Attribute Decision Making. System Engineering and Electronic Technology, 25(5): 517–519.

Part VII
Information Systems Engineering and
Management

An Approach for Prioritizing Agile Practices for Adaptation

Gytenis Mikulenas and Kestutis Kapocius

Abstract Agile software development approaches offer a strong alternative to the traditional plan-driven methodologies that have not been able to warrant successfulness of the software projects. However, the move toward Agile is often hampered by the wealth of alternative practices that are accompanied by numerous success or failure stories. Clearly, the formal methods for choosing most suitable practices are lacking. In this chapter, we present an overview of this problem and propose an approach for prioritization of available practices in accordance to the particular circumstances. The proposal combines ideas from Analytic Hierarchy Process (AHP) decision-making technique, cost-value analysis, and Rule-Description-Practice (RDP) technique. Assumption that such approach could facilitate the Agile adaptation process was supported by the case study of the approach illustrating the process of choosing most suitable Agile practices within a real-life project.

Keywords Agile practices · AHP · Cost-value estimation · Prioritization

1 Introduction

According to the latest CHAOS Report published by Standish Group, still only about 32% of software projects can be called successful, i.e., they reach their goals within a planned budget and on time [31]. In the mid-1990s people started creating new methods because industrial requirements and technologies were moving fast and customers were unable to determine their needs in the early stages of the projects [3, 30]. The year 2001 saw the rise of the Agile development concept. Agile methodologies utilized existing ideas, practices, and methods that were considered incompatible with the traditional plan-driven approaches. Today, there are a lot of

G. Mikulenas (✉)

Department of Information Systems, Kaunas University of Technology, Kaunas, Lithuania
e-mail: gytenis.mikulenas@ktu.lt

the so-called Agile methods that are still evolving, e.g., XP, Scrum, DSDM, Crystal, FDD, ASD, Iconix, OpenUP, and AM.

Although statistics show that traditional methods have not been very successful and despite the availability of numerous alternative approaches, companies tend not to take drastic risks instantly switching from their methodological know-how to the Agile methodologies. Instead, practitioners usually search for ways to introduce more flexibility into the project, thus gradually moving toward Agile development. However, although there is a good choice of Agile methodologies and practices, the formal methods of choosing what fits your needs in terms of cost and value are lacking. The literature on this subject is usually oriented toward describing successful projects and presenting Agile success stories about the application of various techniques in a very specific project environment [12, 13, 16, 25, 27]. This is why the field could seriously benefit from formal and universal ways of learning how to use the Agile methods most efficiently. On the other hand even formal approaches must be adaptable easily enough [32, 43] so that practitioners could apply them.

In this chapter, we discuss the basic applicable parts of Agile methodologies and propose the formal approach for evaluating them using the cost-value analysis. The proposal is based on prioritization of alternatives, which, as we try to show, can lead to the efficient adaptation of available Agile methodologies and practices.

2 Agile Methodologies and Practices

Agile software development is driven by methodologies based on best practices, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams. The term “Agile” was first mentioned in 2001, when the Agile Manifesto was published [3]. Agile methodologies evolved from existing methods, methodologies, and techniques of software engineering and from the successful project stories. The precursor methods and major influences of each of the Agile methodologies are summarized in a map developed by Abrahamsson et al. [2, 14].

According to Situational Method Engineering [17], software development methods can be created by means of construction – identifying small elements of a methodology (variously called fragments, elements, or chunks) and putting them together for a specific situation. Done in an ad hoc manner, this leads to a large number of variations even on one methodology within a single organization [34, 44]. So, what are the core elements in any Agile methodology?

From the theoretical point of view, Agile methodologies could be considered as consisting of the following main elements: stages (phases, builds, milestones), work units (processes, tasks, techniques), producers (people, teams, tools), and work products (documents, models, software items) [14, 19, 36]. But from the Agile point of view, there is no process that fits every project as such, but rather practices should be tailored to suit the needs of individual projects [8]. There is even a new research area that has produced numerous books, articles, and experience reports about the

Table 1 Individual Agile practices

Iterative development	Continuous integration
Regular delivery of working software	Process Miniature
Small releases	Evolutionary design
Configuration management	Flexible architectures
Whiteboard modeling	Collective ownership
Customer tests	Walking Skeleton
Test-driven development	Architectural spikes
Daily stand-ups	Paper-based modeling
Osmotic Communication	Information Radiator
Coding standards	Pair programming
Code refactoring	Easy access to expert users
Simple design	Database Refactoring
Burn charts	User stories

use of individual practices [4, 20]. This shows that these practices are of most interest to practitioners because they can be applied on top of the existing software development methodologies. Some of the best-known individual Agile practices are presented in the Table 1 [1, 2, 4, 10, 20].

3 Agile Adaptation

In the literature, the notion of method adaptation can be referred to by terms such as “method tailoring,” “method fragment adaptation,” or “situational method engineering.” Method tailoring is defined as “A process or capability in which human agents through responsive changes in, and dynamic interplays between contexts, intentions, and method fragments determine a system development approach for a specific project situation.” [7]. Today, one of the biggest challenges in the field of Agile methodologies research is selecting appropriate Agile solutions for a given project. Despite that, most articles on this issue are experience reports describing success stories or lessons learned by organizations that have adopted Agile methodologies for specific projects [12, 13, 16, 25, 27]. These reports are mostly informal and often lack qualitative or quantitative measurements of stated benefits or findings [5].

We analyzed some of the key proposals that are related to the problem of Agile adaptation. The findings are listed in Table 2.

The subject of this chapter is application of Agile practices. Majority of available techniques, however, concentrate on choice making on the methodological level and do not cover the choice of individual components of those methodologies. Clearly there is a shortage of solid analytical approach that would take into account the specifics of the particular situation and would allow formulating a list of appropriately ranked alternatives. However, some of the available solutions could be applied creating such approach. We chose to incorporate the RDP technique despite it being quite basic and only partially meeting our requirements.

Table 2 Proposals for the Agile adaptation

Source and main ideas	Drawbacks
<i>Source [6].</i> New direction in project management success: smart methodology selection. Basically, heavyweight projects should rely on traditional plan-driven methodologies, while Agile solutions should be used in lightweight projects	It is not clear when the project is heavyweight or lightweight. On the other hand, heavyweight projects can benefit from (partial) application of Agile methodologies
<i>Source [9].</i> One of four Crystal methodologies should be chosen for each project depending on the number of people involved and project risks	Only choice of a Crystal family methodology is considered and no partial application is covered
<i>Source [10].</i> Gradual adaptation of methodology. The methodology under consideration should be analyzed in terms of its limitations; practice reflection workshops should be held	Not formal enough. It is not clear when the process can be considered gradual
<i>Source [23].</i> Agile Process Tailoring and problem analysis (APTLY). Suggests that development process should be defined by blending ideas and techniques from best practice and local experience	Critical details of how one should combine various alternatives are missing
<i>Source [31].</i> Proposes using RDP (Rule-Description-Practice) technique: brainstorming Rules of Play and Rules of Engagement	Selection is completely informal, based on brainstorming
<i>Source [35].</i> Advocates assessing and rating project parameters (requirements stability, project size, development timeframe, and others) and analyzing Agile methodologies with respect to their philosophy, process, techniques and tools, scope, outputs, experiences, roles, and responsibilities	Concentrates on assessment and comparison of various methodologies instead of their adaptation; requires deep understanding of Agile methodologies
<i>Source [38].</i> Suggests the construction of an Agile software product-enhancement process by using an Agile Software Solution Framework (ASSF) and situational method engineering	Proposed generic process model is lacking specific tools to make the final decision
<i>Source [37].</i> Proposes using a 4-dimensional analytical tool for measuring adoptability of Agile methodologies	Good understanding of each methodology is required. Oriented toward the comparison of methodologies, while the choice making process is not clear enough.

4 Prioritization

As it was noted before, the question we are facing is how to pick the most suitable Agile practices from the given number of alternative options. Prioritization may be an answer. Majority of prioritization techniques are applied in the field of requirements engineering but these approaches are applicable beyond that. Therefore, in

this research, we used the term “alternatives prioritization” instead of “requirements prioritization” when looking at certain techniques.

One can distinguish a group of techniques that are quite simple and easy to use. For example, the Top-Ten Requirements technique [26] suggests that stakeholders should pick their top-ten alternatives from a larger set without assigning an internal order of the alternatives. This approach is quite suitable when there is difference of opinion between equal stakeholders. Another technique involves assigning each alternative a unique rank from 1 to n [22]. A similar ranking is used in numerical assignment [18], yet here alternatives/requirements are additionally grouped according to their priority (critical, standard, optional, etc.). The problem with these approaches is that it is not possible to see the relative differences between the ranked items, while using loose terms such as critical, standard, and optional could confuse the stakeholders.

Another group includes techniques that are more formal and take into consideration relative importance of the alternatives. The simplest one could be the 100-dollar test prioritization technique [28] where the stakeholders are given 100 imaginary units (money, hours, etc.) to distribute between the alternatives. The main drawback is that there is only one criterion in use. A solid widely accepted technique for decision making is the Analytic Hierarchy Process (AHP) technique [39, 42] according to which all possible alternatives are assessed using a set of criterions. The main advantage is the ability to rank choices according to their effectiveness meeting conflicting objectives. Also, consistency ratios can be applied to test the correctness of the evaluations [41]. AHP prioritization has been applied in approaches such as Quantitative Win-Win [40] or EVOLVE [15].

The bottleneck of AHP in a prioritization scenario is the amount of required pairwise comparisons. The total number of comparisons to perform with AHP are $n \times (n - 1)/2$ (where n is the number of alternatives) at each hierarchy level, which results in a dramatic increase in the number of comparisons as the number of alternatives increases. Researchers have shown that AHP is not suitable for large numbers of requirements [29]. However, this problem can be solved by using two separate criterions – cost and value – and modeling them on a decision-making graph [21]. This way we can simplify AHP at the same time utilizing its strong sides for the ranking of alternatives. The cost criterion reflects resources (material, human, etc.) required to apply the alternative. Value reflects benefits or gains (usability, reuse, automation, etc.) that can be achieved using the given alternative. We believe it is quite sensible to apply these ideas for the prioritization of Agile practices thus facilitating the Agile methodology adaptation. Upcoming sections include details of our proposal as well as the description of the case study.

5 Adapted Prioritization Approach

Proposed adapted prioritization approach incorporates ideas from AHP [41], cost-value [21], and RDP [31] techniques. Majority of Agile methodologies stress the

importance of the project team as a single working, therefore we recommend that all members of the team take part in Agile practices adaptation decisions. The steps of the approach are as follows:

1. *Problems*. Identify the most important or problematic project areas and problems that are in focus.
2. *Suitable Agile practices*. Brainstorm and identify a list of suitable Agile practices, concentrating on practices that fall under Rules of Play or Rules of Engagement categories. Let us suppose the number of suitable practices as n .
3. *Priorities of value*. Calculate priorities of the candidate Agile practices from the common list with respect to the relative value:
 - 3.1 *Pairwise comparison*. Perform pairwise comparisons of the Agile practices according to the criterion of value. The fundamental scale used for this purpose is shown in Table 3.
 - 3.2 *Preparation of the comparison results*. Calculate the total of the n columns for every row in the comparison results matrix. Then divide each matrix element by the sum of the column the element is a part of.
 - 3.3 *Estimation of the priority matrix*. Calculate the sums of each row in the normalized matrix and divide each row sum by the number of candidate Agile practices. The result of this computation is referred to as the priority matrix and is an estimation of the eigenvalues of the matrix.
 - 3.4 *Accuracy check of the estimation results*. Calculate in this order:
 - 3.4.1 *Resulting vector*. Multiply the comparison matrix by the priority vector, then divide each element of the resulting vector by the corresponding element in the priority vector.
 - 3.4.2 $\lambda_{\max}$. It determines the maximum eigenvalue of the comparison matrix and is estimated by calculating the average over the elements in the resulting vector.

Table 3 Scale of pairwise comparisons

Relative intensity	Definition	Explanation
1	Equal value	Two requirements are of equal value
3	Slightly more value	Experience slightly favors one requirement over another
5	Essential or strong value	Experience strongly favors one requirement over another
7	Very strong value	A requirement is strongly favored and its dominance is
9	Extreme value	The evidence favoring one over another is of the highest possible order of affirmation
2, 4, 6, 8	Intermediate values	When compromise is needed between two judgments

Table 4 Random Consistency Index (RI) [41]

<i>n</i>	1	2	3	4	5	6	7	8	9	10
RI	0	0	0.58	0.9	1.12	1.24	1.32	1.41	1.45	1.49

- 3.4.3 *Consistency Index (CI)*. It is the first indicator of the accuracy of the pairwise comparisons and is calculated as $CI = (\lambda_{\max} - n)/(n - 1)$. The closer $\lambda_{\max}$ is to n , the smaller the judgmental errors.
- 3.4.4 *Consistency Ratio (CR)*. It defines the accuracy of the pairwise comparisons and is calculated as $CR = CI/RI$, where RI is a Random Indices value picked up from Table 4. As a general rule, consistency ratio of 0.10 or less is considered acceptable. If it is higher, step back to the pairwise comparison and reconsider the judgments.
- 3.5 *Assignment of priorities*. Assign each Agile practice its priority, i.e., a relative value based on the estimated eigenvalues in the priority matrix.
4. *Priorities of cost*. Calculate priorities of the candidate Agile practices from the common list (perform step 3) with respect to the relative cost.
5. *Cost and value diagram*. Plot the estimated cost and value priorities of the Agile practices on a diagram as the one shown in Fig. 1.
6. *Discussion and choice*. Use the diagram as a conceptual map for selecting Agile practices with higher value-to-cost ratio.

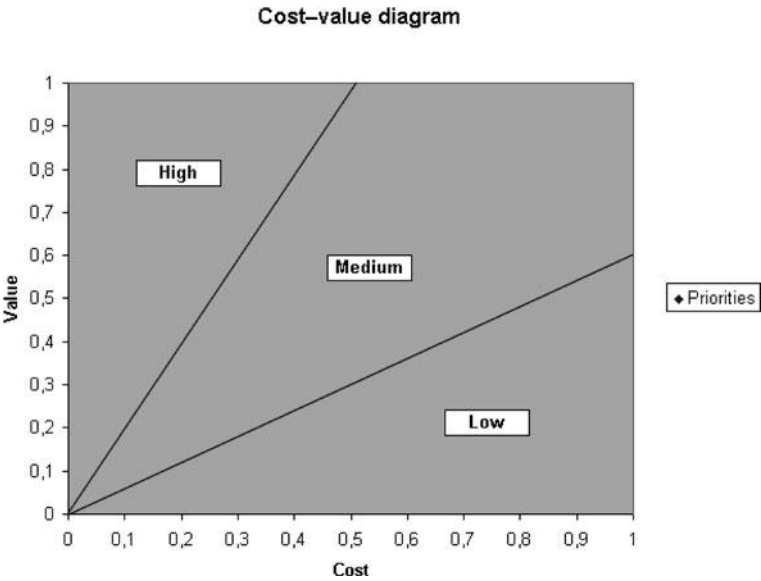


Fig. 1 Value and cost diagram (adapted from [21])

6 Case Study

In 2007–2008, both authors of this chapter took part in the development of the national portal for the stimulation of the practical application of scientific research (www.mokslasverslui.lt). During this project, some of its key features (strict contract, detailed project specification, fixed deadlines, fixed budget) determined that a traditional plan-driven project management was used. However, other characteristics (“invisible” client, unspecific and constantly changing requirements, code quality) resulted in a decision to adapt some Agile practices. A short overview of the process is given below. It is important to stress that although some of the presented calculations may seem complicated, they can be easily automated using spreadsheets (e.g., MS Excel) or performed in specialized tools (e.g., for steps 3 and 4 Expert Choice EC tool could be used).

- [step 1] *Problems*. The problems that were identified are shown in Table 5.
- [step 2] *Suitable Agile practices*. Along with the problems, seven Agile practices (mostly Crystal methodologies [10, 11]) were picked as possibly suitable for our interests (Table 5).
- [step 3] *Priorities of value* and [step 4] *Priorities of cost*. Pairwise comparisons of the selected practices in terms of value and cost (steps 3.1 and 4.1 of the approach) are presented in Tables 6 and 7, respectively. Processed comparison results (steps 3.2, 4.2) are shown in Tables 8 and 9 while the priority matrixes (steps 3.3, 4.3) are shown in Tables 10 and 11.
- [steps 3.4, 4.4] *Accuracy check of the estimation results*. Resulting vector (steps 3.4.1, 4.4.1) for value and cost estimation is given in Tables 12 and 13, respectively. Calculated consistency indexes and consistency ratios (steps 3.4.2–3.4.4, 4.4.2–4.4.4) reveal that pairwise comparisons in terms of both cost and value can be considered accurate (Tables 14 and 15).
- [steps 3.5, 4.5] *Assignment of the priorities*. According to our calculations, the priorities of the Agile practices with respect to value and cost were as follows: Reflective Workshops (0.354 and 0.159, respectively), Osmotic

Table 5 Project problems and suitable Agile practices

Problem	Agile practices
“Invisible” client, unclear requirements	Reflective Workshops (RW)
Non-located team, remote programmers, different work schedules	Osmotic Communication (OC)
Project expansion, staff changes, unused program code, the need for steady portal’s performance	Application Refactoring (AR)
Database Refactoring (DR)	
New technology challenges	Walking Skeleton (WS)
Project status visibility	Information Radiator (IR)
Preservation of accumulated knowledge when project is expanding and staff is changing	Process Miniature (PM)

Table 6 Pairwise comparison (value)

	RW	OC	AR	DR	WS	IR	PM
RW	1	4/1	3/1	2/1	5/1	6/1	7/1
OC	$\frac{1}{4}$	1	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{2}{1}$	$\frac{3}{1}$	$\frac{4}{1}$
AR	$\frac{1}{3}$	$\frac{2}{1}$	1	$\frac{1}{2}$	$\frac{3}{1}$	$\frac{4}{1}$	$\frac{5}{1}$
DR	$\frac{2}{1}$	$\frac{3}{1}$	$\frac{2}{1}$	1	$\frac{4}{1}$	$\frac{5}{1}$	$\frac{6}{1}$
WS	$\frac{1}{5}$	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{4}$	1	$\frac{2}{1}$	$\frac{3}{1}$
IR	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{5}$	$\frac{1}{2}$	1	$\frac{2}{1}$
PM	$\frac{1}{7}$	$\frac{1}{4}$	$\frac{1}{5}$	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{2}$	1

Table 7 Pairwise comparison (cost)

	RW	OC	AR	DR	WS	IR	PM
RW	1	$\frac{2}{1}$	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{3}{1}$	$\frac{5}{1}$	$\frac{4}{1}$
OC	$\frac{1}{2}$	1	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{2}{1}$	$\frac{4}{1}$	$\frac{3}{1}$
AR	$\frac{2}{1}$	$\frac{3}{1}$	1	$\frac{1}{2}$	$\frac{4}{1}$	$\frac{6}{1}$	$\frac{5}{1}$
DR	$\frac{3}{1}$	$\frac{4}{1}$	$\frac{2}{1}$	1	$\frac{5}{1}$	$\frac{7}{1}$	$\frac{6}{1}$
WS	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{5}$	1	$\frac{3}{1}$	$\frac{2}{1}$
IR	$\frac{1}{5}$	$\frac{1}{4}$	$\frac{1}{6}$	$\frac{1}{7}$	$\frac{1}{3}$	1	$\frac{1}{2}$
PM	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{5}$	$\frac{1}{6}$	$\frac{1}{2}$	$\frac{2}{1}$	1

Table 8 Prepared results (value)

	RW	OC	AR	DR	WS	IR	PM
RW	0.386	0.361	0.412	0.449	0.316	0.279	0.25
OC	0.096	0.09	0.069	0.075	0.126	0.14	0.143
AR	0.129	0.18	0.137	0.112	0.189	0.186	0.179
DR	0.193	0.271	0.275	0.225	0.253	0.233	0.214
WS	0.077	0.045	0.046	0.056	0.063	0.093	0.107
IR	0.064	0.03	0.034	0.045	0.032	0.047	0.071
PM	0.055	0.023	0.027	0.037	0.021	0.023	0.036

Table 9 Prepared results (cost)

	RW	OC	AR	DR	WS	IR	PM
RW	0.137	0.18	0.112	0.129	0.189	0.179	0.186
OC	0.069	0.09	0.075	0.096	0.126	0.143	0.14
AR	0.275	0.271	0.225	0.193	0.253	0.214	0.233
DR	0.412	0.361	0.449	0.386	0.316	0.25	0.279
WS	0.046	0.045	0.056	0.077	0.063	0.107	0.093
IR	0.027	0.023	0.037	0.055	0.021	0.036	0.023
PM	0.034	0.03	0.045	0.064	0.032	0.071	0.047

Table 10 Priority matrix (value)

E1	E2	E3	E4	E5	E6	E7
0.354	0.104	0.159	0.24	0.068	0.045	0.031

Table 11 Priority matrix (cost)

E1	E2	E3	E4	E5	E6	E7
0.159	0.106	0.237	0.35	0.07	0.032	0.046

Table 12 Resulting vector (value)

R1	R2	R3	R4	R5	R6	R7
7.341	7.169	7.286	7.359	7.073	7.049	7.105

Table 13 Resulting vector (cost)

R1	R2	R3	R4	R5	R6	R7
7.286	7.169	7.359	7.341	7.073	7.105	7.049

Table 14 Accuracy results (value)

$\lambda_{\max}$	CI	CR
7.197	0.033	0.037

Table 15 Accuracy results (cost)

$\lambda_{\max}$	CI	CR
7.21	0.035	0.038

Communication (0.104 and 0.106), Application Refactoring (0.159 and 0.237), Database Refactoring (0.24 and 0.35), Walking Skeleton (0.068 and 0.07), Information Radiator (0.045 and 0.032), and Process Miniature (0.031 and 0.046).

[step 5] *Cost and value diagram.* Aforementioned values plotted on a cost-value diagram are shown in Fig. 2.

[step 6] *Discussion and choice.* Estimation results reflected our concerns and interests. The top winning Agile practice was Reflection Workshops. We rejected Process Miniature practice as its value-to-cost ratio was very low. Database and Application Refactoring practices were quite tempting as they had second and third highest evaluations. However, their high price meant that the resulting value-to-cost ratio was not very attractive and this was why these techniques were not adopted. In other words, we chose to perform the Refactoring in a more traditional and controlled manner instead of doing it whenever needed. On the other hand, the findings pushed us toward adopting Osmotic Communication, Walking Skeleton, and Information Radiator practices, because of their reasonable value-to-cost ratio.

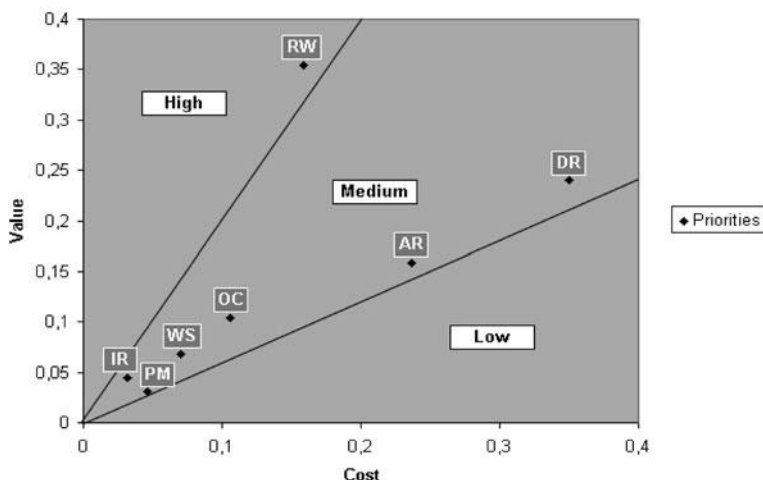


Fig. 2 Value and cost diagram of the Agile practices

7 Conclusions

There is a wealth of available practices that can be applied along with various Agile methodologies. However, the issue of selection of the most appropriate practices has not been well researched and majority of approaches are based on making informal empirical choices. We have proposed the formal accessible approach that allows companies to prioritize available practices in accordance to the particular circumstances and use this list when making the decisions about the Agile development process.

The proposal incorporates ideas from AHP, cost-value evaluation, and RDP techniques. The main advantage of this approach is its ability to evaluate and rank Agile practices according to their comparative value and cost. There is also a way to formally test the evaluations for consistency and accuracy.

However, no formal method can always provide the best answer. That is why it is essential that final decisions on which practices to adopt are made collectively by project team members using the cost-value diagram (step 6 of the approach).

The contribution is threefold. First, analysis of available research revealed that although Agile methodologies consist of various types of elements, often only individual practices are being applied. Second, it was shown that the proposed approach of Agile practices prioritization can be applied adapting such practices. And finally, the use of cost and value criteria for the comparative estimation of alternatives is a practical way of evaluating Agile practices and it can noticeably facilitate the move toward the truly more Agile development.

Acknowledgment This work is supported by the Lithuanian State Science and Studies Foundation according to the High Technology Development Program Project “VeTIS” (Reg. No. B-07042).

References

1. Abrahamsson, P., Salo, O., Ronkainen, J., and Warsta, J. (2002) Agile Software Development Methods: Review and Analysis, VTT Publications.
2. Abrahamsson, P., Warsta, J., Siponen, M. K., and Ronkainen, J. (2003) New directions on Agile method: A comparative analysis. In: Proceedings of the 25th International Conference on Software Engineering, IEEE Computer Society, pp. 244–254.
3. Agile Alliance (2001). Principles behind the Agile Manifesto. Retrieved 14 May, 2009, from: <http://agilemanifesto.org/principles.html>.
4. Ambler S. W. (2007) Agile Adoption Rate Survey: March 2007. Retrieved 15 May, 2009, from: <http://www.ambysoft.com/downloads/surveys/AgileAdoption2007.ppt>.
5. Ambu, W. and Gianneschi, F. (2003) Extreme programming at work. In: M. Marchesi and G. Succi (Eds.), 4th International Conference on Extreme Programming and Agile Processes in Software Engineering, XP 2003. Lecture Notes in Computer Science, Berlin: Springer, Vol. 2675, pp. 298–306.
6. Attarzadeh, I. and Hock, O. S. (2008) New direction in project management success: Base on smart methodology selection. In: Proceedings of Information Technology Symposium 2008, Vol. 1, pp. 1–9.
7. Aydin, M. N., Harmsen, F., Slooten, K. V., and Stagwee, R. A. (2004) An Agile Information Systems Development Method in Use. Turkish Journal of Electrical Engineering, 12(2): 127–138.
8. Beck K. (2004) Extreme Programming Explained: Embrace Change, 2nd Edition. Addison Wesley Professional.
9. Cockburn, A. (2000) Selecting a Project's Methodology. IEEE Software, IEEE Computer Society Press, Vol. 7(4), pp. 64–71.
10. Cockburn A. (2004) Crystal Clear: A Human-Powered Methodology for Small Teams. Addison Wesley Professional.
11. Cockburn, A. (2006) Agile Software Development: The Cooperative Game, 2nd Edition. Addison Wesley Professional.
12. Drobka, J., Noftz, D., and Raghu, R. (2004) Piloting XP on Four Mission-Critical Projects, IEEE Computer, IEEE Computer Society Press, Vol. 21(6), pp. 70–75.
13. Elssamadisy, A. (2001) XP on a Large Project – A Developer's View, In Proceedings of XP/Agile Universe, Raleigh, NC.
14. Firesmith D.G. and Henderson-Sellers, B. (2002) The OPEN Process Framework. An Introduction. London, Addison-Wesley.
15. Greer, D. and Ruhe, G. (2004) Software release planning: An evolutionary and iterative approach. Information and Software Technology 46(4): 243–253.
16. Grenning, J. (2001) Launching Extreme Programming at a Process-Intensive Company, IEEE Software, pp. 27–33.
17. Henderson-Sellers, B., Gonzalez-Perez, C., and Ralyte, J. (2008) Comparison of Method Chunks and Method Fragments for Situational Method Engineering. Software Engineering ASWEC 2008, IEEE Computer Society, Vol. 18(6), pp. 479–488.
18. IEEE Std 830-1998 (1998) IEEE recommended practice for software requirements specifications. IEEE Computer Society, Los Alamitos.
19. ISO/IEC. (2007) ISO/IEC 24744, Software Engineering. Metamodel for Development Methodologies. International Standards Organization/International Electrotechnical Commission.
20. Jeffries, R. E., Anderson, A., and Hendrickson, C. (2000) Extreme Programming Installed, Addison-Wesley.
21. Karlsson, J. and Ryan, K. (1997) A cost-value approach for prioritizing requirements. IEEE Software 14(5): 67–74.
22. Karlsson, J., Wohlin, C., and Regnell, B. (1998) An evaluation of methods for prioritizing software requirements. Information and Software Technology 39(14–15): 939–947.

23. Keenan, F. (2004) Agile Process Tailoring and problem analysis (APTLy), In: The Proceedings of the 26th International Conference on Software Engineering, pp. 45–47.
24. Kroll, P. and MacIsaac, B. (2006) *Agility and Discipline Made Easy: Practices from OpenUP and RUP*. Addison Wesley Professional.
25. Lan, C., Mohan, K., Peng, X., and Ramesh, B. (2004) How Extreme Does Extreme Programming Have to Be? Adapting XP Practices to Large-Scale Projects. In: Proceedings of the 37th Hawaii International Conference on System Sciences, IEEE Press, Vol. 3, pp. 342–250.
26. Lausen, S. (2002) *Software Requirements – Styles and Techniques*. Pearson Education, Essex.
27. Layman, L., Williams, L., and Cunningham, L. (2004) Exploring extreme programming in context: An industrial case study. In: Proceedings of the Agile Development Conference, IEEE Computer Society, pp. 32–41.
28. Leffingwell, D. and Widrig, D. (2000) *Managing Software Requirements – A Unified Approach*. Addison-Wesley, Upper Saddle River.
29. Lehtola, L. and Kauppinen, M. (2004) Empirical evaluation of two requirements prioritization methods in product development projects. In: Proceedings of the European Software Process Improvement Conference (EuroSPI 2004), Springer, Berlin, Heidelberg, pp. 161–170.
30. Lindvall, M., Basili, V., Boehm, B., Costa, P., Dangle, K., Shull, F., Tesoriero, R., Williams, L., and Zekowitz, M. (2002) Empirical findings in Agile methods. *Agile Universe*, pp. 197–207.
31. Lynch, J. (2009) New Standish Group report shows more project failing and less successful projects. Press release. Standish Group, Boston, MA, Retrieved 21 May, 2009 from: http://www.standishgroup.com/newsroom/chaos_2009.php.
32. Mikulenas, G. and Butleris, R. (2009) An approach for modeling technique selection criterions. In: The Proceedings of the 15th International Conference on Information and Software Technologies, IT 2009, Kaunas University of Technology, pp. 207–216.
33. Mirakhorli, M., Rad, A.K., Aliee, F.S., Mirakhorli, A., and Pazoki, M. (2008) RDP technique: Take a different look at XP for adoption. *Software Engineering*, In: The Proceedings of the ASWEC Conference, pp. 656–662.
34. Mirbel, I. (2006) Method chunk federation. In: The Proceedings of the 18th Conference on Advanced Information Systems Engineering, Namur University Press, pp. 407–418.
35. Mnkandla, E. and Dwolatzky, B. (2007) Agile methodologies selection toolbox. *Software Engineering Advances, ICSEA*, pp. 72–72.
36. OMG (2002). *Software Process Engineering Metamodel Specification*, formal/2002-11-14. Object Management Group.
37. Qumer, A. and Henderson-Sellers, B. (2006) Measuring agility and adoptability of Agile methods: A 4-dimensional analytical tool. *IADIS International Conference Applied Computing*, pp. 503–507.
38. Qumer, A. and Henderson-Sellers, B. (2007) Construction of an Agile Software Product-Enhancement Process by Using an Agile Software Solution Framework (ASSF) and Situational Method Engineering. *Computer Software and Applications Conference COMPSAC 2007*, Vol. 1, pp. 539–542.
39. Regnell, B., Höst, M., Natt och Dag, J., Beremark, P., and Hjelm, T. (2001) An industrial case study on distributed prioritization in market-driven requirements engineering for packaged software. *Requirements Engineering* 6(1): 51–62.
40. Ruhe, G., Eberlein, A., and Pfahl, D. (2002) Quantitative WinWin – A new method for decision support in requirements negotiation. In: Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering (SEKE'02), ACM Press, New York, pp. 159–166.
41. Saaty, T. L. (2000) *Fundamentals of the Analytic Hierarchy Process*. RWS Publications.
42. Saaty, T. L. (2007) Multi-decisions decision-making: In addition to wheeling and dealing, our national political bodies need a formal approach for prioritization. *Mathematical and Computer Modelling* 46(7–8): 1001–1016.

43. Silingas, D. and Butleris, R. (2008) Towards implementing a framework for modeling software requirements in MagicDraw UML. *Information Technology and Control*, Kaunas, Technologija, 38(2): 153–164.
44. Tutkute L., Butleris R., and Skersys T. (2008) An approach for the formation of leverage coefficients-based recommendations in social network. *Information Technology and Control*, 37(3): 245–254.

Effects of Early User-Testing on Software Quality – Experiences from a Case Study

John Sören Pettersson and Jenny Nilsson

Abstract It is a well-established fact that the usability of a software package increases if it has been tested with users and usability flaws have been corrected. However, this presentation does not focus on usability issues but on the effects of early user-testing on software code quality determined as the number of errors discovered in function tests. In analysing the results of an extensive software update cycle, the authors found that functional requirements based on the results of early user-testing resulted in program code that had half the number of the errors (and less than one-fifth of the critical errors) found in code based solely on requirements emanating from users' verbal opinions.

Keywords Software quality · User-centred software development · Early user-testing · Wizard-of-Oz prototyping

1 Introduction

User-centred software development has not been primarily interested in “testing” per se but rather in collaborative work methods that strive to include users in the process of defining the scope and aims of a software development project. This is generally acknowledged as a prominent trait of the “Scandinavian tradition” [1], but is also a general topic within the usability community [2]. Typical methods are “anthropological” observations at work places to see what people really are doing, interviews with people, and focus groups to discuss work routines and possible impacts of and demands on new software. To some extent, early user-testing is mentioned in the participatory design literature, referring to methods such as paper prototyping [3], scenario-driven approaches [4] and card sorting exercises [5].

J.S. Pettersson (✉)

Department of Information Systems, Karlstad University, Karlstad, Sweden
e-mail: john_soren.pettersson@kau.se

Naturally, there is a tug-of-war between developers and usability experts when it comes to the question of how much effort should be put into refining usability. Admittedly, early user-testing entails a cost for developing the mock-ups, a work that programmers think they will cover anyhow as they are developing the product; “just wait a few months and you can test a preliminary version of the real product”. As is well known from the criticism launched by the “Agile” movement, e.g. [6], this preliminary version tends to be a limping monstrosity which often cannot be used for realistic user tests, and it is not delivered on time anyhow. Not even teams using the agile methods necessarily gear towards user experience [7]. Moreover, the cost of the work that is normally put into correcting errors and omissions in software that are in fact caused by incomplete specifications should also be included in the total assessment of the perceived added cost of different types of user involvement in the design process. Workload cannot only be measured in person-hours spent, but one must also consider the different salary costs for mock-up makers-and-testers and system programmers in such total assessments. There is, of course, the further question of the cost at the user side if usability flaws are not corrected or if the workflow is not correctly aligned with the actual demands of the work.

However, in this study we want to focus on the non-usability effects of usability work, and we will not pit different early user-testing methods against each other, but rather discuss how and to what degree early user-testing improves software quality as compared to other, more verbally oriented user-inclusive approaches. Our interest lies in methods that provide detailed “statements” about the possible appearance of future software systems, because this results in early feedback on interaction design issues and also on the functions that the intended user groups will require.

The case study presented in this chapter concerns an update cycle of a large software package, RIB, at the Swedish Rescue Services Agency (more on this in Section 3). This software update, which one of the authors (Nilsson) was working with at the time, included a complete reprogramming of the four largest modules. To ensure usability, early user-testing was performed using the so-called Wizard-of-Oz method. This method entails that a test user interacts with a realistic mock-up, where the output on the test user’s “computer” (monitor, loudspeaker, or whatever physical user interface is used) is generated by the test leader. This realistic look and feel of the mock-up makes it possible to test not only the intelligibility of individual graphical design ideas or particular naming of labels but also realistic workflows where the demands of the resulting work process can be explored. This provides test users with a hands-on experience of the consequences and possibilities behind the demands they have given voice to in preliminary interviews, questionnaires, and perhaps also jointly in focus groups.

The mock-ups that are a part of this chapter’s case study included clickable maps and other things that might have been laborious to set up even in the form of a faked computer prototype if the experimenter had had to program a Wizard-of-Oz environment that could then be used to *avoid* programming the prototypes themselves. However, our research group has developed a general Wizard-of-Oz system with which experimenters can piece together a mock-up using graphics, whole scenes, and details of graphical user interfaces, and then add the controls and behaviours

allowing the test leader to control the interactive aspects of the mock-up during each test session (the test leader is a so-called “Wizard of Oz”; [8]). We call this system *Ozlab* [9]. *Ozlab* has been used repeatedly during a 3-year period to make mock-ups of some of the modules in the software package of our case study. More on the use of this system will follow in Section 3.

The chapter starts off by elaborating our main thesis and describing the method used to assess the difference in code quality between two approaches for early user involvement (Section 2). Upon this follows a description of the software package (Section 3.1) in the case study, the use tests (Section 3.2), and the debugging process (Section 3.3). Our analysis is broken down into three main parts: different stakeholders’ involvement in mock-up testing (Section 4.1), error rates in different program modules in the case study, and a final comparison between two modules (Section 4.3). Section 5 summarises our main findings and discusses the specific method we used for the mock-up testing.

2 Main Thesis

This chapter has as its leading thesis that the quality of software, from a software engineering viewpoint, may be increased if the development process complements verbal forms of early requirements gathering with early user-testing. That is, performing user tests where designers’ solutions to user requirements are concretised with interactive illustrations not only increases usability (which to some extent can be mended during programming by testing the system under development) but also improves the code’s quality. The reasoning behind our thesis is:

- Extending the system specification phase from simple, verbal forms of “user participation” to also include usability testing almost always entails further needs specification. This leads to a more complete specification before programming takes place, which in turn results in less re-programming during the development phase.
- Specifications derived solely from user-voiced requirements are less precise and are therefore more prone to be reformulated by the ordering organisation, the designers, and the programmers as the development proceeds. This leads to more errors in the code and a certain “spaghetti” trait as the changes are made after the overall architecture has been set.

That is, we argue, in essence, for more complete specifications¹ which may sound like a version of the much criticised Water Fall model [10] to some

¹The need to express systems’ requirements in a form understandable by various user groups has been raised by several authors, including our team when developing our *Ozlab* tool and *Ozlab* methodology (especially [11]). Here, however, we will not speak about increasing users’ ability to voice requirements and find alternative solutions.

readers. However, our approach includes letting interaction designers test whether specifications are complete with respect to the interactivity they are to support.

2.1 Method

The question is how to evaluate code quality in order to make it possible to ascertain the existence of a difference between the two early user-involvement methods: early user-testing vs. more conscious inputs from the user groups. The task was to assess code quality when it was *either* programmed based on mock-upped and user-tested designs, initially made from perceived needs by real users, *or* programmed only according to perceived needs by real users.

The quality of the code for each program module was gauged as the number of errors (found in a debugging process described in Section 3.3) divided by the size of the program or divided by the number of files belonging to that program. The analysis below is based on both types of ratios.

It should be understood that it has not been possible to directly calculate statistical significances for the different programs in this software package. We will discuss the differences in error rates because different parts were error-prone to different extents. There was also the question if individual files or whole programs should be investigated; the latter was finally selected (as will be explained below), but this meant that there was only one example of a program based on non-pre-tested code (of sizeable updating and extension). We also compare differences in the number of new requirements.

3 Case Study Description

As previously mentioned, the case study concerns an extensive update of a large software package in the area of decision support system for civil protection. Certain parts of this package have been updated through carefully executed use tests based on mock-ups in order to make certain that function specifications and graphic designs have been made clear and unambiguous before programming starts. Other parts of the software package have been updated with requirements gathered from the large user group and implemented by an external software developer. Before the update was released, all parts went through a final debugging process conducted by three evaluation groups consisting of experienced users, the RIB HCI expert and the content experts, all of whom are experts on the functional requirements of the system.

3.1 Description of the Software Package

The software package, RIB, is an integrated decision support system for civil protection, developed by MSB, The Swedish Civil Contingencies Agency (new agency

from 1 January 2009; during the case study RIB was developed by The Swedish Rescue Services Agency).

RIB is a source of information for everyone who works with civil protection, i.e. fire-fighters, police officers, medical personnel and coastguard officers, hauliers, and municipal civil servants. It combines various databases that provide comprehensive information about how to deal with an emergency as well as details about how prevention work can be planned, the risks involved once an emergency has happened, and where to find resources for the emergency response. The main parts and the parts that have been considered in this study are the following:

The Library contains close to 15,000 items. These items include research reports, experience reports, fire investigations, training literature, legislation, and also films and Internet links.

Hazardous Substances contains physical facts about a large number of substances, e.g. boiling point, melting point, vaporisation point, and flammability range. This part also contains information about experiences gained during emergencies involving the substance, as well as contact details for experts and information about prevailing legislation.

Resources contains information about resources that can be used during large emergencies and emergency response operations. The resources, in the form of material and expertise, are available at fire brigades, businesses, organisations, and authorities, and their locations are displayed on a map.

Operational Support allows the user to organise and categorise his/her own operational documents, and, during operations status, register events, decisions, manpower, tactics and trends in order to obtain an overview of the operation.

In addition to the main parts, the software package also contains an overall search of RIB, *Statistics* (which now have been removed from the program), and approximately 20 smaller tools and training programs.

3.2 Use Tests

During the development of The Library, Hazardous Substances, and Resources, two use tests were conducted using the so-called Wizard-of-Oz method and the Ozlab system. The reason for using Wizard-of-Oz and other methods for early user-testing is that the alternative way, that is, programming prototypes and then testing these, is expensive. Programming is a time-consuming and costly task [12], and bugs that make it impossible to run user tests at all must be removed before user-testing can start, which will delay testing. Consequently, user-testing will not be made early and when it is done, there is little willingness to throw the prototypes away and start over again since so much effort will have been expended on the prototype's programming. On the other hand, a Wizard-of-Oz-based test also includes an up-front effort since the mock-up must be constructed: one simply must have graphics and texts in the mock-up's user interface for display on the test user's monitor in order to test anything at all. However, using the WOz method, one can refine the faked prototype and conduct user tests throughout this refinement process.

The interactive mock-ups in the case study were produced in two steps. In step one, the images of the graphical user interface were created and the interactivity planned. In the second step the interactivity support for the “Wizard” (i.e. test leader) was added to the images by dragging and dropping pre-defined Ozlab behaviours to the graphical elements [13]. These behaviours make it possible for the test leader to control the interactive aspects of the mock-up during test sessions.

Use testing was conducted in two rounds: one on a rough design, with eight test participants, and one 6 months later on a detailed design, with five new test participants. When creating the rough design, very little work was put into making the mock-up look final; instead, it was supposed to look sketchy. The detailed design, on the other hand, was made to resemble a final design. Even though the level of detail in the two prototypes differed, both test sessions evaluated the intelligibility of the designs, including the naming of labels and buttons, and the general workflow. The test users were selected from the target group of the software package and were asked to solve a number of tasks using the mock-up while thinking out loud. The interaction was observed and each test session was followed by a questionnaire.

The second mock-up went through some minor changes after testing, e.g. a design company delivered specifications on text fonts and colour schemes, etc. It was then used as a base for the functional and graphical specification of the program modules. In addition, the programmers had to interact with the mock-ups before starting to program the interaction in the modules.

3.3 The Debugging Process

The debugging process commenced nearly a year before the final launch of the new version. The debugging was conducted by three groups: very experienced users, the HCI expert (the co-author of this chapter), and the persons responsible for the different databases in the modules. All three of these groups are experts on the functional requirements of all or some of the parts of the total system. The experienced users were all from the target group and had extensive experience with previous versions of the software package.

Table 1 gives the total number of unique errors, reported by the three evaluator groups, in the Early User-Tested (EUT) programs and the no-EUT program. The errors are presented according to the classification made by the RIB HCI expert; priority 1 being the most serious errors that had to be fixed, and priority 3 being the least serious errors that did not need to be fixed before release (priority 3 was in the error lists noted as “Next version”, and was often new requirements). The fourth column of error types, “Content”, contains content-related errors in databases, texts (including spelling errors), and pictures.

Note that the bug-finding by experienced users also resulted in new requirements. This was also the case for the debugging made by the content experts, who had not been involved in the pre-tests before programming; they had only seen and accepted the requirements specifications.

Table 1 Errors found in the debugging process before the release of the new version

Program	Prio 1	Prio 2	Prio 3 nxt	Content	Not errors
The Library (early user-tested)	30	119	24	7	9
Resources (EUT)	26	95	4	3	6
Hazardous Substances (EUT)	7	98	4	13	2
Operational support (no-EUT)	65	189	13	6	4

4 Analysis

There are several indications on the benefits of the early user-testing approach. In the analysis of error rates in Section 4.3 (below) only two of the four program modules in the case study are used. Naturally, we will have to argue for why we pick only two programs for the comparison. First, however, we will make a note on the need to include the development team in the pre-tests.

4.1 Improve Requirements Specifications by Letting Developers Conduct Tests

The Ozlab system does not only entail early user-testing, but is also a way of communicating between different stakeholders in a software development process. This was however not used (fully) in the RIB development cycle described above. The content experts, that is, the persons responsible for the different modules’ databases, belong to the development team even if they are not programmers. Thus, they are not users of the RIB system, and this fact meant that they were not invited to the Ozlab sessions for the three early user-tested program modules (The Library, Resources, and Hazardous Substances). However, they partook in the final debugging as one of the three groups of expert evaluators. This meant that the librarian, responsible for the The Library, found several reasons to make requirements in the error report, as reflected in Table 2.

While the higher number of new requirements of Operational Support is not surprising given that no early user-testing was conducted for Operational Support (this fact simply underlines our main thesis), the numbers for The Library is

Table 2 New requirements identified in the debugging process

Program	Number of new requirements	Size (MB)	Number of files
The Library (early user-tested)	24	0.7	55
Resources (EUT)	4	0.7	57
Hazardous Substances (EUT)	4	1.5	145
Operational support	13	2.0	230

more remarkable. It should be understood that the developers, including the content experts, had gone through the requirements specifications before development started. They were all happy with the specifications and tentative screen layouts. Analysing the exact nature of the new requirements for The Library reveals that the high number of new requirements (“unproportional” if compared to the program size and the number of constituent files) could have been considerably lowered if the content expert in question had interacted with the mock-ups. That is, even without making special mock-ups or tasks for the expert evaluator (the librarian), the tests could have yielded these requirements if only this expert had been included among the participants.

4.2 Error Rates in Different Program Modules of RIB

Turning now to the general picture of errors in the four main program modules in our post hoc analysis of code quality, it turns out that the two smaller modules (The Library and Resources) have a high number of prio 1 and prio 2 errors. Both were subjected to early user-testing, so this seems to be counter to our main thesis. On the other hand, as the error rates are much higher than the error rates for the early user-tested module Hazardous Substances as well as for the not pre-tested module Operational Support, the cause is certainly not early user-testing. Furthermore, as The Library and Resources are only one-third in size and in number of files of Hazardous Substances, one would have expected these two modules to be simpler and less complex (counted for instance in McCabe Metric [14] or some Functional Size Measurement [15]). A closer look reveals the following:

- The Library relies heavily on database references throughout its code, and therefore the possibility for the programmer to make severe errors are much higher than for other modules of RIB. Also included in the error count are new requirements which were plenty for The Library as noted in Section 4.1.
- In Resources, the users report directly to the database, which means that classifications are often problematic – to compare, the databases of The Library and Hazardous Substances are built by the content experts, and for Operational Support, where the users are supposed to build up their own document database, a simple sample database was used in this update cycle of RIB.
- The last fact pertaining to Operational Support should in fact make the updating of this program less error-prone compared to the updating of Hazardous Substances. In order to make a fair comparison between the two programs, one could consider developing a scaling factor for the error rates. However, for the purpose of testing our main hypothesis this is not a problem because unscaled figures should speak less in favour of our hypothesis.

In conclusion, of the early user-tested program modules, it is only Hazardous Substances that should be included in the comparison with the not early user-tested module, Operational Support.

Of the four programmers, each working as the lead programmer for one of the four modules, three were in-house and the one working with Operational Support was an external consultant. This could possibly have had an influence on error rates. However, considering the number of errors detected in the two smaller EUT modules, the in-house/external dimension does not seem to play a role (for the possible positive effect of the external programmer on the “Content” errors, see next paragraph). It should be noted that debugging was equally extensive for all four modules, and the provenance of the code had no influence on debugging and reporting routines.

The column “Content” in Table 1 contains the number of errors which were not really dependent on the *program modules* and *database structures* but related to errors in the content. From the table it can be seen that there were fewer in the not pre-tested Operational Support than in the three early user-tested programs (scale for sizes as in Table 2). The third point above explains partly why there were few content errors in Operational Support: there was little content to err in. Furthermore, content errors are not easy to spot during user-testing unless the most experienced users (and other content experts) participate. This was the case in the debugging process but not in the early user-testing process. We exclude the fourth column from most of the comparative discussion below.

4.3 Error Rates in Comparison

In order to calculate the error rates for the individual program modules (in this presentation, Hazardous Substances and Operational Support), some sort of normalising operation had to be applied. One approach could be to normalise the size of the programs, as large programs would tend to have more errors. Complexity tends to increase dramatically with increasing number of code lines. For the same reason, one can expect that the number of files of a program may increase the complexity. In fact, we have used both figures for normalisation.

An alternative to compare the two modules on module level would have been to compare individual files. In this way one would get a high number of comparisons and it would be possible to consider doing statistical analyses on the significance level of the differences in error rates that exist between the two modules. However, the lesson from the error rate analysis above of The Library and Resources is that one has to be very clear of the inherent problems of a module in order to be able to judge whether or not it can be compared with another module. Indeed, the programmers had already pointed out Hazardous Substances as the module that was most like Operational Support for our comparison of EUT vs. no-EUT. Picking out individual files for a statistical analysis would introduce a great number of uncertainties. In conclusion, this kind of large software update cannot be replicated in contrast to laboratory experiments where “all other things being equal” provides the grounds for exact statistical analyses.

On the aggregated level that the two modules have to be compared we spot a clear difference, as revealed in Table 3.

Table 3 Error rates relative to program size (MB and # of files)

	Error type Program					
	Prio 1		Prio 2 + 3		Priority 1,2,3	
	#/MB	#/files	#/MB	#/files	#/MB	#/files
Hazardous Substances (EUT)	4.67	0.05	68.00	0.70	72.67	0.75
Operational support (no-EUT)	32.50	0.28	101.00	0.88	133.50	1.16
Error rates proportionally (EUT/no-EUT)	0.14	0.17	0.67	0.80	0.54	0.65

Note: Priority 3 was in the report lists noted as “Next version”, often new requirements.

Table 3 shows that the program code based on functional requirements which were based on early user-testing had half the number of errors (and less than one-fifth critical errors) compared to code that was based on functional requirements based solely on requirements from user groups.

When including the Content errors, the EUT-developed module rate is 81 and 0.84 while the no-EUT module’s rates are 136.50 and 1.19, which makes the proportion of errors in the EUT module relative to the errors in the no-EUT to become as high as 0.60 and 0.71 based on module sizes and module files, respectively. This does not make the advantage of EUT to appear as prominent as the figure in Table 3, but from the discussion ending the previous section we argue that these proportions do not show what really is the gain of early user-testing.

5 Conclusions

Lessons learnt from this case study on the net effect of early user-testing, EUT, on code quality (not counting usability quality) can be summarised as:

1. decreases critical errors,
2. decreases overall error rate and
3. includes developers.

In addition, it should be noted that *more* requirements, not only more *complete* requirements, are found through the early user-testing procedure. If this procedure includes content experts as well, *even* more requirements are identified; the general concept is *early use(r)-testing*; cf. [16].

Naturally, “explanations”, as presented in Section 2, to motivate our thesis could always be questioned if no direct relationships can be established. Here, we must introduce another kind of caveat, namely the quality of the mock-ups being tested in early use-testing. The quality might be decisive for how much new information one may get as compared to focus groups and other methods of requirements gathering from user groups. In this case study, very complete interaction designs were

presented to the users, especially as concerns the interaction, less complete as concerns the graphics in the first test cycle. This is a procedure which entails not only certain “extra” costs but also a demand of a skilled “Wizard” for setting up interactive mock-ups in the Ozlab system and conducting the individual test sessions. Fortunately, such a Wizard existed, as she had extensive experience working with Ozlab from assisting research projects at Karlstad University [17] and, due to her university studies in Multimedia, skilled in Photoshop. It is interesting to note that salary-wise there seems to have been no problem for the employer to motivate the elaboration of pre-tests (indeed, the agency producing RIB is hiring her on a regular basis after she was recruited back to the university in January 2009).

Admittedly, it would have been useful to compare with other early user-involving, design-testing methods, as well as comparing early user-testing (that is, testing before specifications are finalised) against different degrees of agility in systems development processes (perhaps to optimise mixed methods, [6]). As it is, the data for this study was present after the process was completed. When the error probing was done, the feeling emerged that although a lot of interaction design issues had been fixed during development, Operational Support suffered on another level, namely in the code structure itself. Of course the question arises of what to compare with. Such a large update cycle as that of RIB is not cheap to run doubled as two comparable processes, one with and one without EUT. As Section 4.2 made clear, it is not meaningful to compare just any two module developments. The comparison in this study has been made with careful consideration of such difficulties and the results are quite unequivocally speaking for the hypothesis that early user-testing brings not only usability into software but also reduces programming errors, possibly through avoiding that fixes are added because such fixes are likely to go against the general architecture of the software.

References

1. Ehn, P. *Work-Oriented Design of Computer Artifacts*. Lawrence Erlbaum Assoc. Hillsdale, NJ, USA (1990).
2. ISO 13407:1999 *Human-Centred Design Processes for Interactive Systems*. ISO/TR 18529:2000 *Ergonomics – Ergonomics of Human-System Interaction – Human-Centred Lifecycle Process Descriptions*. International Organization for Standardization (1999 & 2000).
3. Snyder, C. *Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces*. Morgan Kaufmann, San Francisco, CA, USA (2003).
4. Carroll, J. *Making Use: Scenario-Based Design of Human-Computer Interactions*. The MIT Press, Cambridge, MA, USA (2000).
5. Spencer, D. and Warfel, T. Card sorting: A definitive guide. http://www.bboxesandarrrows.com/view/card_sorting_a_definitive_guide (2004).
6. Agile manifesto. <http://agilemanifesto.org/> (2001).
7. Ambler, S.W. Tailoring usability into agile software development projects, in *Maturing Usability: Quality in Software, Interaction, and Value*, Chapter 4, eds. E.L.-C. Law, E. Hvannberg and G. Cockton. (Human-Computer Interaction Series) Springer, London (2008).

8. Kelley, J.F., An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Office Information Systems* 2(1): 26–41 (1984).
9. Pettersson, J.S. Visualising interactive graphics design for testing with users. *Digital Creativity* 13(3): 144–156 (2002).
10. Royce, W.W. Managing the development of large software systems. In WESCON 70 (Los Angeles, Aug. 25–28, 1970). IEEE, San Francisco, CA, 1970, pp. 1–9. Reprinted in 9th International Conference on Software Engineering (Monterey, California, March 30–April 2, 1987). Los Alamitos, CA, USA: IEEE Computer Society Press, pp. 328–338 (1987).
11. Molin, L. and Pettersson, J.S. How should interactive media be discussed for successful requirements engineering? in *Perspectives on Multimedia: Communication, Media and Technology*, eds. Burnett, Brunström and Nilsson. Wiley, Chichester (2003).
12. Cooper, A. *The Inmates Are Running the Asylum: Why High-Tech Products Drive Us Crazy and How to Restore the Sanity*. Indianapolis, IN: Sams, Cop. (1999).
13. Nilsson, J. and Siponen, J. Challenging the HCI concept of fidelity by positioning Ozlab prototypes, in *Advances in Information Systems Development*, eds. A.G. Nilsson et al., Springer, New York (2006).
14. McCabe, Th.J. A Complexity Measure, *IEEE Transactions on Software Engineering*, SE-2(4): 308–320 (1976).
15. ISO/IEC 14143-1:1998 (rev. 2007) *Information Technology – Software Measurement – Functional Size Measurement – Part 1: Definition of Concepts*. International Organization for Standardization (1998 & 2007).
16. Molin, L. Wizard-of-Oz prototyping for cooperative interaction design of graphical user interfaces, in *Proceedings of the 3rd Nordic Conference on Human-Computer Interaction*, 23–27 October, Tampere, Finland, pp. 425–428 (2004).
17. PRIME project, Privacy and Identity Management for Europe, a 6FP EU project; usability work reported in deliverable series D6.1a-d. www.prime-project.eu.

Development of Watch Schedule Using Rules Approach

Darius Jurkevicius and Olegas Vasilecas

Abstract The software for schedule creation and optimization solves a difficult, important and practical problem. The proposed solution is an online employee portal where administrator users can create and manage watch schedules and employee requests. Each employee can login with his/her own account and see his/her assignments, manage requests, etc. Employees set as administrators can perform the employee scheduling online, manage requests, etc. This scheduling software allows users not only to see the initial and optimized watch schedule in a simple and understandable form, but also to create special rules and criteria and input their business. The system using rules automatically will generate watch schedule.

Keywords Scheduling · Software design · Online software · Business modelling

1 Introduction

There are many tools used for schedule creation. But these tools are pointed to create watch schedules irrespective of the additional occupation of staff. These tools are used when the work is organized on on duty basis.

Some institutions organize additional work involving being on duty, for example, army (Lithuanian Military Academy), where training is running together with arming. Sometimes monthly on duty schedules are needed; in that case the monthly additional occupation of staff must be taken into account. This data must be picked up and analyzed and after that schedules are created.

One of the ways to create a schedule taking into account these limitations is to use rules. Rules can be used to describe the status of employees.

D. Jurkevicius (✉)

Department of Information Systems, Faculty of Fundamental Sciences, Vilnius Gediminas Technical University, Vilnius, Lithuania
e-mail: d.jurkevicius@isl.vgtu.lt

Other parts of this chapter consist of these sections:

- The related software.
- Formal concept analysis and its usage are described in the first section.
- Multiple (distributional) formal context is described in the second section.
- The method of creating watch schedule is described in the third section.
- The experiment is described in the fourth section.
- Conclusions are presented in the last section.

2 The Related Software

Some scheduling online programs were analyzed (Table 1): Watch schedules manager (our created program), ClockIt-Online, myShiftManager, WhenToWork, and

Table 1 The scheduling online software

	Duty schedules manager	ClockIt- online	myShift- Manager	WhenTo- Work	Work schedule DOT NET
Automatically create schedules	X	X	X	X	X
Finding the optimal decision by availability criteria for generating schedule	X				
Generating schedule by availability		X	X	X	X
Editing of schedule and choosing other alternative	X	X	X	X	X
Possibility to create the same watch schedules	X	X	X	X	X
Set availability	X	X	X	X	X
Set availability by criteria (weights of criteria)	X				
Using rules for generating schedule	X				
Billboard	X	X	X	X	X
Send report for the user		X	X	X	X
Company month calendar	X	X	X		X
Time sheet	X	X		X	X
Agenda	X	X	X	X	X
Employee handling	X	X	X	X	X
Manage shift definitions	X	X	X	X	X
Manage department definitions	X	X	X	X	X

Work schedule DOT NET. All other tools except watch schedules manager are used for organizing work in shifts or duties. The watch schedules manager is used for organizing work just in duties.

3 Understanding of Formal Concept Analysis

One of the ways to transform available data in hierarchic aspect is formal concept analysis. Dau [1] noticed that scientists in making plots could not lean on them as on arguments. To separate formally the mathematical structure from its scheme presentment the work environment was created in which diagrams could be used to make formal substantiations. Here we define some terms used in this chapter.

Concept can be defined as

- an abstract or general idea inferred or derived from specific instances [12];
- a concept is an abstract idea or a mental symbol, typically associated with a corresponding representation in language or symbology, that denotes all of the objects in a given category or class of entities, interactions, phenomena, or relationships between them [2];
- it has an intention (deep definition), extension (set of objects or exemplars) [7]; and
- the definition of a type of objects or events. A concept has an intensional definition (a generalization that states membership criteria) and an extension (the set of its instances) [8].

Formal Concept Analysis (FCA) [10] method is

- a mathematization of the philosophical understanding of concept;
- a human-centered method to structure and analyze data; and
- a method to visualize data and its inherent structures, implications, and dependencies.

FCA is based on the philosophical understanding that a concept can be described by its extension – that is all the objects that belong to the concept and its intension which are all the attributes that the objects have in common [9].

Formal context – the mathematical structure used to formally describe these tables of crosses (or briefly a context) [11].

FCA is a method used in data analysis, knowledge imaging, and information control. Rudolf Wille suggested FCA in 1981 [10] and it is successfully developed nowadays. For 10 years FCA was researched by small groups of scientists and Rudolf Wille's students in Germany. FCA was not known worldwide because the biggest part of publications was presented at mathematicians' conferences. After getting the sponsorship, some projects were implemented in this area. Most of them were knowledge research projects used in work. This system was known only in

A	B	C	D	E	F	G	H	I	J
	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9
Obj 1									
Obj 2	X								
Obj 3									
Obj 4									
Obj 5									
Obj 6									
Obj 7									
Obj 8									
Obj 9									
Obj 10									
Obj 11									
Obj 12									
Obj 13									
Obj 14									
Obj 15									

Fig. 1 Example of formal context

Germany. During the last 10 years FCA became the research object of international scientists' community. FCA was used in linguistics, psychology, also in projecting the software, in the areas of artificial intelligence and information search.

Some of the structures of FCA appeared to be fundamental to information representation and were independently discovered by different researchers, for example, Godin et al.'s [4] use of concept lattices (which they call "Galois lattices") in information retrieval.

Here we introduce the formal concept analysis definition [3].

For example, G is the set of objects that we are able to identify in some domain (e.g., if, when, than). Let M be the set of attributes. We identify the index I as a binary relationship between two sets G and M , i.e., $I \subseteq G \times M$. A triple (G, M, I) is called a formal context. For $A \subseteq G$, we define

$$A' := \{m \in M \mid (g, m) \in I \text{ for all } g \in A\} \quad (1)$$

and dually, for $B \subseteq M$

$$B' := \{g \in G \mid (g, m) \in I \text{ for all } m \in B\} \quad (2)$$

A formal concept of a formal context (G, M, I) (Fig. 1) is defined as a pair (A, B) with $A \subseteq G$, $B \subseteq M$, $A' \subseteq B$, and $B' \subseteq A$. Sets A and B are called extend and intend of the formal concept, respectively. The set of all formal concepts of a context (G, M, I) is called the concept lattice of context (G, M, I) .

4 Distributional Formal Contexts

In this chapter we propose [6] the distribution of the traditional formal context (Fig. 1) in three parts:

- objects are described in the first table (objects);
- attributes are described in the second table (attributes); and

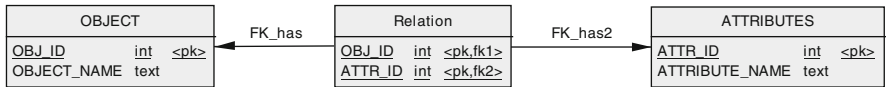


Fig. 2 Proposed logical scheme and notation of distributional (extended) formal context

- relations between objects and attributes are described in the third table (relation). Proposed physical data model is shown in Fig. 2.

But traditional context (Fig. 1) is needed for formal concept analysis. The additional transformation is needed for converting distributional formal context to traditional context. We propose the transform table where relations between objects and attributes are saved.

That transformation is used for formal concept analysis. Relations of table transformations into traditional context are shown in Fig. 3. The tradition formal context is created after transformation.

That solution allows connecting to distributional formal context additional data. Additional data can be either traditional formal context or distributional formal context. The type of context depends on one reason. The distributional formal context is always parent. Relationships between contexts are generally represented in context rectangles by a line with dot. Context is parent when line ends with dot.

Logical data model, where the relation between two distributional formal contexts is represented, is shown in Fig. 4.

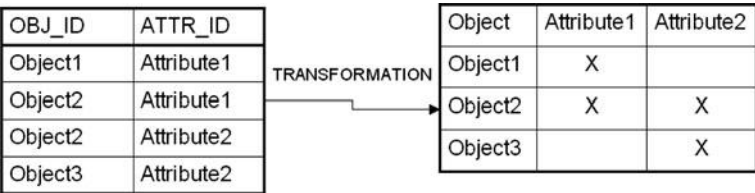


Fig. 3 Transformation of distributional formal context into traditional formal context

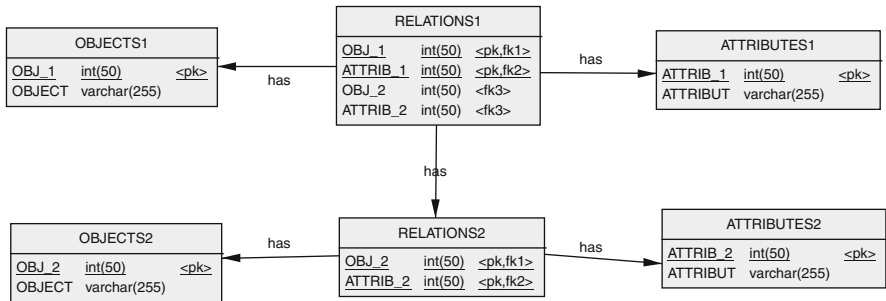


Fig. 4 Relationship between two distributional formal contexts

Our proposed method is deriving *formal concepts* from *distributional formal context* with the following routine (first method when context 1 is main and used for data analysis and context 2 will be used for saved additional data):

1. pick a set of objects A from context 1;
2. derive the attributes A' from context 1;
3. derive all the formal concepts B' from context 2;
4. derive $(A')'$ from context 1;
5. derive all the formal concepts B'' from context 2; and
6. (B'', B') is a *formal concept*.

5 The Algorithm of Creating of Schedule

Now we will describe proposed method how to create watch schedule using formal concepts analysis for data analyze and rules.

The main idea of creating watch schedule is to find the biggest conflicts objects (people) and attributes (number of days). The intersection of these is the biggest conflict that we had found. Then the number of the founded conflicts will reduce.

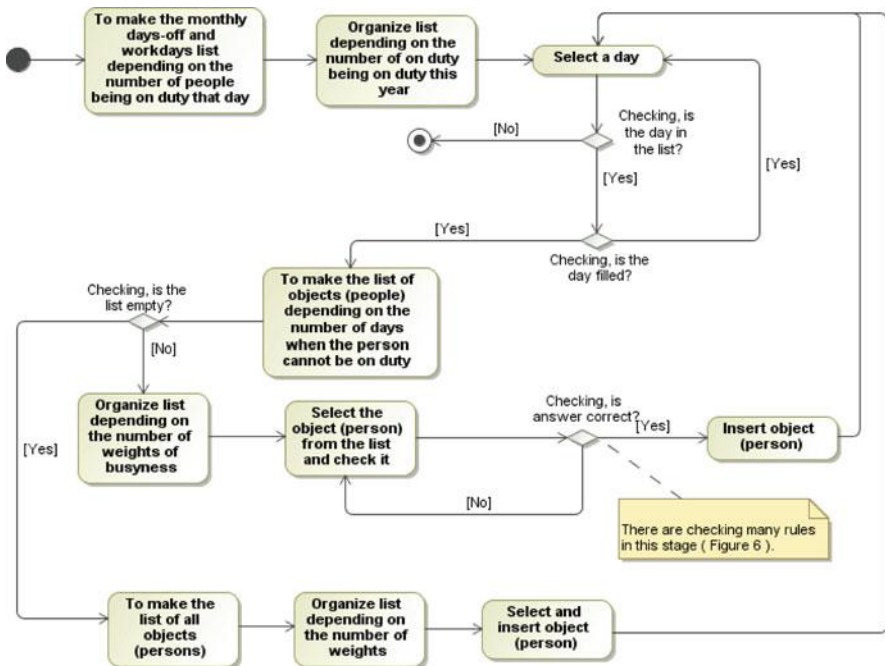


Fig. 5 Algorithm of schedule

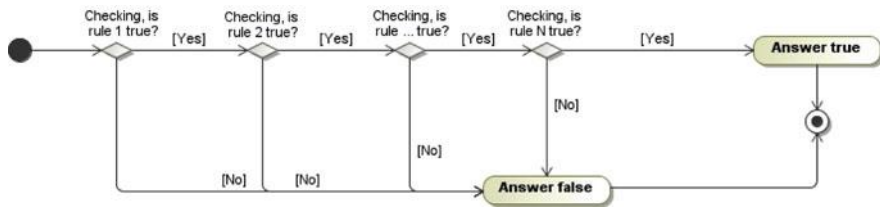


Fig. 6 Explanation of how to check the objects using rules

The schedule's algorithm (Fig. 5):

1. to make the monthly day-off list (using FCA) depending on the number of people being on duty that day (Fig. 6a);
2. to make the monthly list (using FCA) work-days depending on the number of people being on duty that day (Fig. 6b);
3. organize list depending on the number of on duty being on duty this year;
4. select the day;
5. if there are free days in the list then go to 8;
6. if there are no free days in the list then go to 17;
7. make the list (using FCA) of objects (people) depending on the number of days when the person cannot be on duty (Fig. 7);
8. if the list of objects is empty then go to 14;
9. if the list of objects is not empty then go to 10;
10. organize list (using FCA) depending on the number of weights of business;
11. select the object (person) from the list and check it (Figs. 6 and 8);
12. if this can be labeled then insert and go to point 4;
13. if this cannot be labeled then go to point 11;
14. make the list of all objects;
15. organize list (using FCA) depending on the number of weights of business;
16. select and insert first object from list and go to 4; and
17. the end.

The rules checking process is shown in dotted area of Fig. 5. This process is explained in Fig. 6.

For example, after data analysis (Fig. 7a, b) we got the sequence of attributes (days): 2, 1, 8, 9 (day-off) and 6, 5, 4, 3, 7 (work-day). Other data analysis is shown in Fig. 8, where k is attribute (day) numbers. For example, for the day 4 we got the sequence of objects (persons): 5, 6, 7, 9, 13, 14. Then these objects are being analyzed and checked. This stage is shown in Fig. 9. The object 5 wins because its final weight was lower (get 0) than other. Then, the object (winner) is inserted into the database (as an example (Fig. 9) – object 5 shall be inserted into the database).

Figures 6 and 9 show the example how to check objects. One stage is rules, and second stage is weights checking. Sources of rules are personal section, study

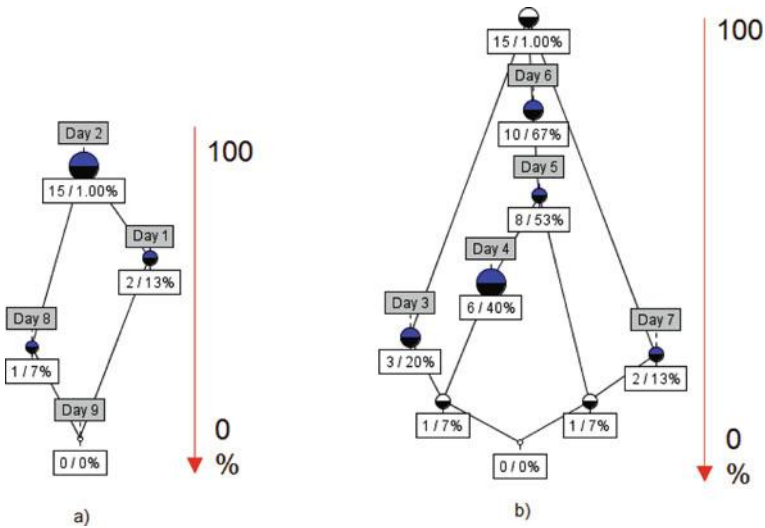


Fig. 7 Generation of concepts lattice and their organizing according attributes. (a) According to day-off; (b) according to work-day

schedule, etc. There is a reason to use rules machine (for holding rules) because some kind of rules change quite quickly, for example, rules in natural language:

- person (John) cannot go on duty from January to February and
- people holding certain appointments can go on duty just during day-off.

When the user had described rules, then these rules can be ambiguous and without definition [5]. We can remove indefinites and ambiguity, if each rule is resolved into elementary or atomic rules. Atomic business rule is declaratively written, using natural language, easily understudied by businessmen, and it is not ambiguous. Information system designers write atomic rules using formal language. During this stage of rules transformation cross-purposes can occur because users have their own language and system creators have their own. Mistakes occurring during the stage of the rules transformation process can be removed if a user writes the declarative rule by the template proposed in natural language.

Making the rule input in the declarative form, it is suggested to make the input using the template or this rule can be written using semistructured natural language.

Using the template input proposed by the system the initial component of the rule (e.g., If) is suggested [5]. Then other terms that are kept in the formal context are suggested. It is a step-by-step constructed rule. The mistake can be avoided using the first method, when the templates are suggested, and the rule can be immediately transformed into the formal form. This method is designed mainly for the businessmen. Hereunder we show that different rules have generic form. Rules can

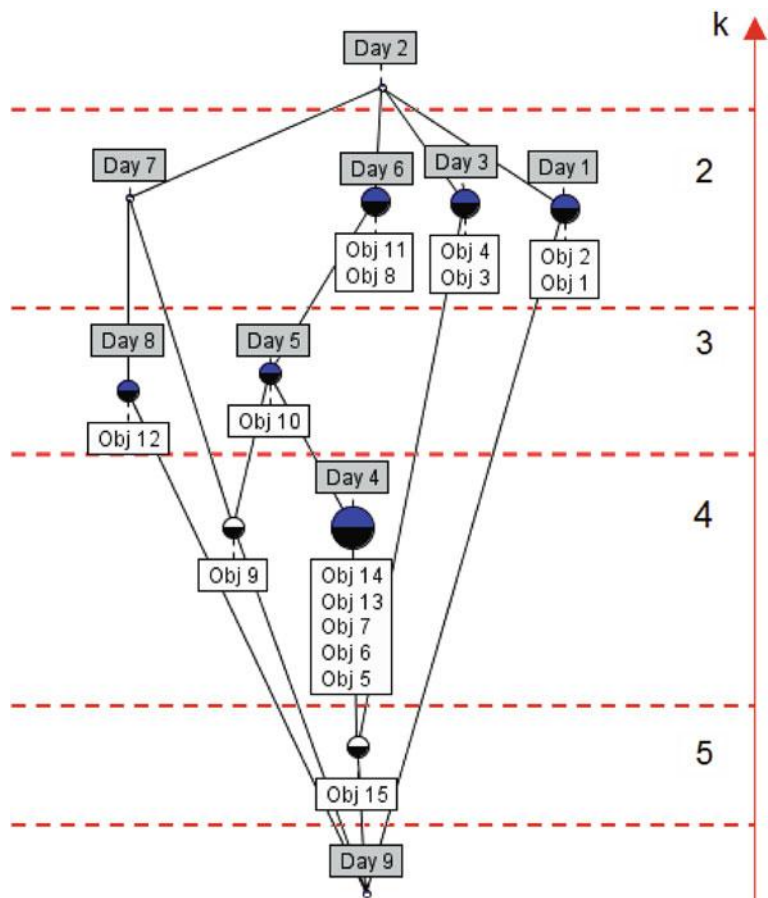


Fig. 8 Generation of concepts lattice and their organizing according objects

Fig. 9 Example of objects analysis (the best object 5 is selected)

All objects belong to day 4 (one formal concept)						
Objects:	5	6	7	9	13	14
Result of checking the rules:	1	1	0	1	0	0
Weights:	0	1	-	5	-	-

be defined by declarative IF-THEN form with the priority, start date, finish date, for example:

- First rule:
- If person is John and month is January or February then reject.

Second rule:

- If person is studying and seeking the master degree and now is a work-day then reject.

At the first sight these rules do not have much common. After analyzing the rules it was recognized that what condition and action of the rule can be made from some components.

If we write this rule in the most common way, it would be

If condition, then action.

There are not so many ways to formulate the condition and action.

6 Experiment

A tool (prototype) has been created to create the watch schedules and manage them. This tool, a web application, is called the Duty Schedules Manager (DSM) (Fig. 10). The prototype is being tested in Lithuanian Military Academy.

DSM has a complete feature set to enable you to create your own schedule.

Main features of DSM are

- generating the schedule automatically;
- finding the optimal decision by availability criteria for generating schedule;
- editing of schedule and choosing other alternative;
- possibility to create the same watch schedules;

Ed. Nr.	Vardas, pavardė	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	ALL			
																																		9	2008	2008	R
1	Plt. Lm. Valdemaras Rupsiys												R			R			X			X											X	2	0	2	2
2	Mjr. Eugenijus Karpas		X																R					X				R						2	1	2	2
3	Mjr. Svajūnas Papilkauskas												X				X			R			R											2	0	2	2
4	Mjr. G. Jakšlys				R			R			X																				X			2	0	2	2
5	Mjr. I. Balūnas				R												R					X					X							2	1	2	2
6	Mjr. Darius Labanauskas			X		R																			X				R					2	1	2	2
7	Kpt. T. Eismontas			R														X			R					X								2	1	2	2
8	Kpt. Virginijus Jonuška					X			R														R					X						2	1	2	2
9	Kpt. Darius Medūnas							X																			R			X				2	1	2	2
10	Kpt. Vytytis Andreika				X							R		X		R															R			2	0	2	2
11	Kpt. Ausėnas Buškus													R					X			X				R								2	0	2	2
12	Kpt. Ramantas Zukas						X			R																	R			X				2	1	2	2
13	Kpt. Ernestas Jociuskas			X				R					X																	R		R	2	0	2	2	
14	Vr. Dmitrij Mackovskij				R							X			X					R											R			2	0	2	2
15	Darius Jurkevicius							X				R																			X			2	1	2	2

Fig. 10 The application for creating watch schedule

- user can get his/her own data about availability;
- billboard;
- send e-mail;
- duty roster: online scheduling;
- time sheet;
- employee handling;
- manage shift definitions;
- manage department definitions; etc.

7 Conclusions

The duty schedules manager is used for organizing work just in duties. By using the proposed method, the user is allowed to get his/her data about availability. The schedule is generated more precisely than other analyzed online scheduling software, because it uses availability criteria of users. Our future plans are to explore how to make rules manages easier.

Acknowledgments The work is supported by Lithuanian State Science and Studies Foundation according to High Technology Development Program Project “Business Rules Solutions for Information Systems Development (VeTIS)”, Reg. No. B-07042.

References

1. Dau F. (2004). Types and tokens for logic with diagrams. In: K. E. Wolff, H. Pfeiffer, H. Delugach (eds), *Conceptual Structures at Work. Proceedings of 12th International Conference on Conceptual Structures*. Springer, Berlin, pp 62–93.
2. ENCYCLOPEDIA Britannica (2009). *Concept. The Classic Encyclopedia Based on 1911 Edition of the Encyclopedia Britannica*. <http://www.1911encyclopedia.org/Concept> [Accessed on 2009-07-18].
3. Ganter B., & Wille. R. (1999). *Formal Concept Analysis: Mathematical Foundations*. Springer, Berlin.
4. Godin R., Gecsei J., & Pichet C. (1989). Design of browsing interface for information retrieval. In: N. J. Belkin, & C. J. van Rijsbergen (eds), *Proceedings of the SIGIR '89*, pp 32–39.
5. Jurkevičius D., & Vasilecas O. (2008). Rules transformation using formal concept approach. In: Papadopoulos, G. A., Wojtkowski, W., Wojtkowski, W. G., Wrycza, S., Zupancic, J. (eds), *Proceedings of the 17th International Conference on Information Systems Development (ISD2008)*. Information Systems Development: Towards a Service Provision Society, Springer, New York.
6. Jurkevičius D., & Vasilecas O. (2009). Formal concept analysis for concepts collecting and their analysis. Computer Science and Information Technologies, Latvian University, Riga, Nr. 751 (2009), pp 24–41.
7. Martin J., & Odell J. (1994). *Object-Oriented Methods: A Foundation*. Prentice-Hall, p 52.
8. Mayers A., & Maulsby D. (2004) Glossary <http://acypher.com/wwid/BackMatter/Glossary.html> [Accessed on 2007-12-11].

9. Tilley T. (2003). *Formal Concept Analysis Application to requirements engineering and design*. <http://www.int.gu.edu.au/~intille/publications/tilley04formal.pdf> [Accessed on 2007-11-15].
10. Wille R. (1982). Restructuring Lattice Theory: an approach based on hierarchies of concept. In: I. Rival (ed), *Ordered Sets*. Reidel, Dordrecht-Boston, pp 445–470.
11. Wolf K. (1993). A first course in formal concept analysis. In: Faulbaum, F. (ed.) *SoftStat'93 Advances in Statistical Software 4*, pp 429–438.
12. WORDNET (2008) *A lexical database for the English language*. <http://wordnet.princeton.edu/perl/webwn?s=concept> [Accessed on 2008-01-08].

Priority-Based Constraint Management in Software Process Instantiation

Peter Killisperger, Markus Stumptner, Georg Peters, and Thomas Stückl

Abstract In order to reuse software processes for a spectrum of projects, they are described in a generic way. Due to the uniqueness of software development, processes have to be adapted to project-specific needs to be effectively applicable in projects. This instantiation still lacks standardization and tool support making it error-prone, time consuming, and thus expensive. Siemens AG has started research projects aiming to improve software process-related activities. Part of these efforts has been the development of an architecture for a system executing instantiation decisions made by humans which automatically restores correctness of the resulting process.

Keywords Software process · Constraint · Instantiation

1 Introduction

Explicitly defined software processes for the development of software are used by most large organizations. At Siemens AG, business units define software processes within a company-wide Siemens Process Framework (SPF) [20] by using semiformal Event-Driven Process Chains (EPC) and Function Allocation Diagrams (FAD) [19]. They are not defined for projects individually but in a generic way as reference processes for application in any software project of the particular business unit.

Due to the individuality of software development, reference processes have to be instantiated to be applicable in projects. That is, the generic description of the process is specialized and adapted to the needs of a particular project. Until now, reference processes are used as general guideline and are instantiated only minimally

P. Killisperger (✉)

Competence Center Information Systems, University of Applied Sciences, München, Germany
e-mail: peter.killisperger@siemens.com

by manual creation of project-specific plans at Siemens. A more far-reaching tool-supported instantiation is desirable, because manual instantiation is error-prone, time consuming, and expensive. The main reason is the complexity of processes and the need to comply with modeling constraints. The former mainly derives from the hierarchical structure of the processes and their size of several thousand entities. The latter (i.e., constraints) are part of the SPF.

For improving current practice, a Software Engineering Framework (SEF) [16] has been defined. The integral part of the SEF is gradual instantiation of software processes. Here we define *instantiation* as tailoring, resource allocation, and customization of artifacts.

The area of project-specific composition and adaptation of software processes and methods has attracted significant attention in recent years as in, e.g., Brinkkemper's Method Engineering (ME) proposal [10] as an approach for the creation of situational methods. However, no existing approach provides a thorough solution for instantiating Siemens processes. For example, Brinkkemper's situational method configuration process emphasized bottom-up assembly of project-specific methods from fragments, requiring very detailed fragment specifications. Contrary to ME, approaches like SLANG [4] regard processes as programs, enacted by machines [11]. Here, however, we are concerned with flexible method engineering in the large and deal with semiformal process models offering high-level guidance for humans.

Existing tools provide only minimal support for instantiation. Decisions made by humans have to be executed mostly manually. For example, Rational Method Composer (RMC) [13] allows changes on software processes, but although approaches have been developed making the user aware of inconsistencies caused by instantiation operations [14], the actual correction of the process is still left to humans.

For example, consider an activity $a1$ connected by a control-flow to an activity $a2$ which in turn is connected by a control-flow to an activity $a3$ ($a1 \rightarrow a2 \rightarrow a3$). A project manager wants to delete $a2$ because the activity is not needed in the particular project at hand. He selects $a2$ and removes it from the process. Additionally, he has to take care of restoring correctness, i.e., has to make sure that the adapted process complies with modeling constraints defined by his organization. For example, he has to establish the broken control-flow between $a1$ and $a3$ and has to take care of affected information-flows and resource connections.

In order to noticeably reduce the considerable effort for instantiation, tool support has to be extended. Our goal is to derive a flexible architecture for systems that execute instantiation decisions made by humans and automatically restore correctness of the resulting process. We define a process to be *correct* when it complies with the constraints on the process defined in a method manual. A method manual is a meta-model defining permitted elements and constructs, derived from restrictions of the used process definition language and organizational restrictions (e.g., SPF).

The chapter is structured as follows: Section 2 introduces the SEF and describes the developed architecture for instantiation of software processes. It then introduces the correction approach used in the architecture, before particular design decisions

are discussed. Section 3 contains an evaluation of the approach. Related work is discussed in Section 4, followed by conclusions.

2 Priority-Based Constraint Management

2.1 Software Engineering Framework

On the basis of information collected in interviews with practitioners at Siemens AG, a Software Engineering Framework (SEF) (Fig. 1) for improving the instantiation and application of software processes has been developed [16].

The SEF consists of a *reference process*, gradual instantiation by *high-level* and *detailed instantiation*, and an *implementation of the instantiated process*. *High-level instantiation* is a first step toward a project-specific software process by adapting the *reference process* on the basis of project characteristics and information that can already be defined at the start of a project and are unlikely to change. Such characteristics can be, e.g., the size of a project (a small project will only use a subset of the process) or required high reliability of the software product which requires certain activities to be added.

High-level instantiation is followed by *detailed instantiation* which is run frequently during the project for the upcoming activities. A step-by-step approach is proposed, because it is often unrealistic to completely define a project-specific process already at the start of a project [6].

The resulting instantiated process can be used in projects in different ways including visualization of the process and management of project artifacts.

Although instantiation in the SEF is split into two distinct stages, it is advantageous if both are based on the same principles. A set of elemental *Basic Instantiation Operations* (BIO) have been defined which are used for both stages [15]. Examples are: “Deleting an activity” or “associating a resource with an activity.” In *high-level*

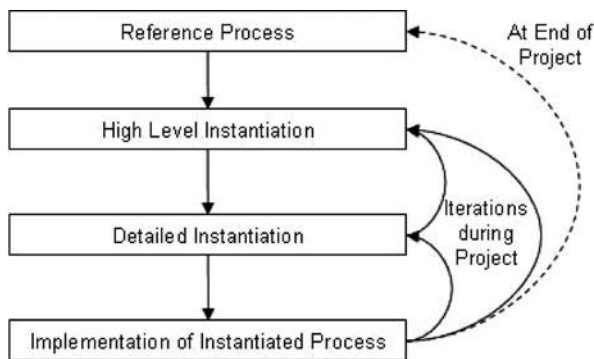


Fig. 1 New Software Engineering Framework

instantiation, BIOs are executed on process elements as batch (i.e., predefined process adaptations depending on, e.g., the project type) and in *detailed instantiation* individually. Using the same principles enables flexible definition and adaptation of predefined instantiation batches.

Existing tools (as mentioned in the introduction) allow performing those changes but they do not guarantee adherence of modeling constraints. That is, current tools do not automatically correct a process when constraints on the process are violated due to instantiation decisions and their execution.

Constraints are derived from process modeling languages and organizational modeling policies such as the SPF. Examples are: “A manual activity has to have at least one human participant executing the activity” or “an artifact has to be created by an activity before it can be input of an activity.”

In order to reduce the effort of instantiation and to guarantee adherence of constraints we have developed an architecture for executing instantiation decisions made by humans which automatically restores correctness of the resulting process. The way a process is instantiated and what constraints are to be met by the process depends on the applying organization and on the used process modeling language. The architecture of an instantiation system has to support the following:

- Differing method manuals since organizations impose different constraints their processes have to meet. Constraints might also change over time.
- Differing BIOs since organizations instantiate their processes in a different way. Organizations might also change the way they instantiate their processes over time.

By taking into account the requirements stated above, the architecture described in Fig. 2 has been developed.

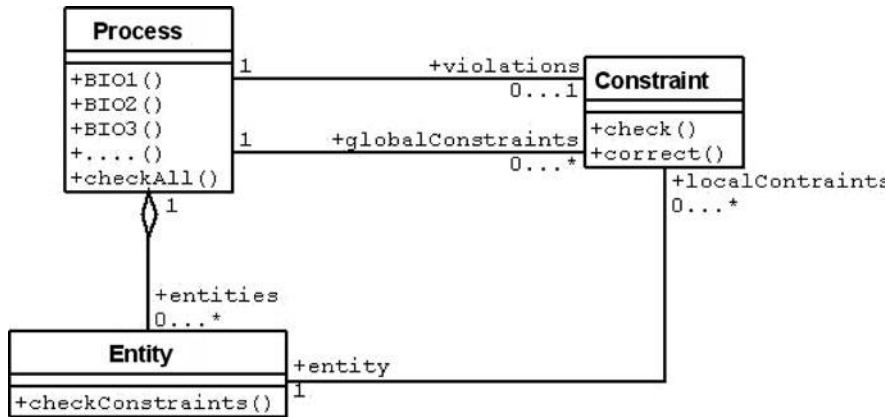


Fig. 2 Class diagram for system for instantiation of software processes

2.2 Architecture Framework

A process consists of different types of entities (e.g., phases, activities, control-flows, and resources). These types can differ depending on the meta-model of the process used by the organization. A process and its entities can have constraints restricting their properties and relationships with other entities. Constraints can be classified into

- Constraints restricting a type of an entity (local constraints), e.g., an activity must have exactly one outgoing control-flow.
- Constraints restricting a process (global constraints), e.g., a process must have exactly one start-event.

A process can be instantiated by running Basic Instantiation Operations (BIO1(), BIO2(), ...) on it which adapt a process in a predefined manner. The resulting process might violate constraints. Constraints are checked (checkall()) and when at least one constraint is violated, its method correct() is called. Correct() adjusts the process so that it complies with the violated constraint. The procedure of checking constraints and executing correct() of violated constraints is continued until all constraints are satisfied. The final process is handed back to the user for further instantiation.

2.3 Correction Approach

How a violation is to be corrected depends on the environment in which the entities causing the violation are settled in the process[0]. It might be possible to correct a violation not only in one particular way but in a variety of ways depending on properties and relationships of entities in the process. The way a violated constraint is corrected also affects what following violations occur and how they can be corrected. Consider the following example: A process consists of the entities start-event, three activities (A1, A2, A3), and an end-event (Fig. 3a).

The project manager of a project decides to adapt the process by running the BIO “inserting a parallel/alternative path” which is defined as follows:

The user selects two control-flows from where the new path diverges/rejoins and specifies the type of split and join for diversion/rejoin. Both control-flows are deleted by the system and a split and a join of the chosen type is created.

The decision of the project manager to run the BIO on the process of Fig. 3a results in the incorrect process of Fig. 3b. The control-flow connecting A1 and A2 as well as the control-flow connecting A2 and A3 have been removed. An XorSplit and an XorJoin have been created.

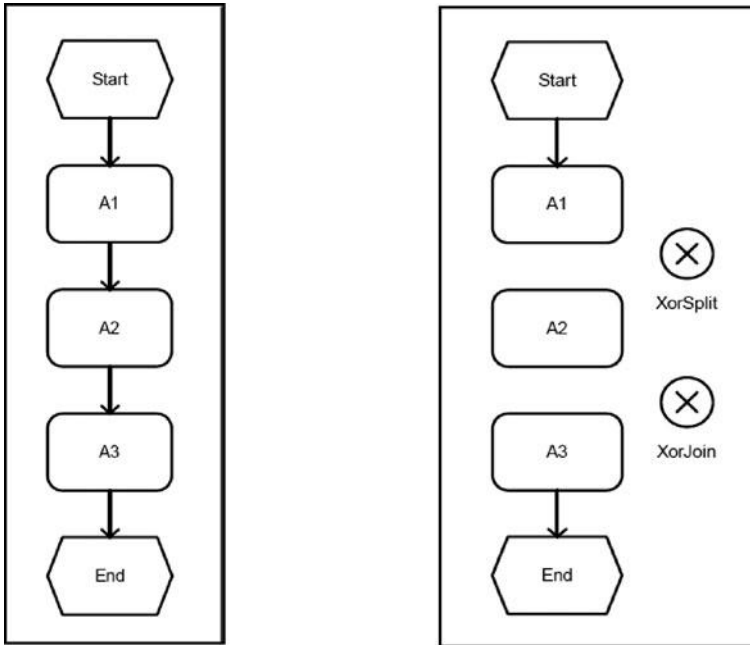


Fig. 3 (a) Example instance of process. (b) Result after executing BIO “Inserting a parallel/alternative Path”

The example process itself and its entities are restricted by a number of constraints:

- A1, A2, and XorJoin have to have one outgoing control-flow.
- A2, A3, and XorSplit have to have one incoming control-flow.
- XorSplit has to have at least two outgoing control-flows.
- XorJoin has to have at least two incoming control-flows.

Each constraint has a method `correct()` specifying how to adapt the process in order to satisfy the constraint. For example, the rule associated with the violated constraint that A1 has to have one outgoing control-flow says that a new control-flow leading from A1 to an entity with too few incoming control-flows is created.

In the context of A1 in Fig. 3b there is more than one possibility to correct the process since there are several entities having too few incoming control-flows. A new control-flow to A2, A3, XorJoin, or XorSplit can be created. The decision which entities are connected influences the further correction procedure (i.e., has an impact on what resulting correct process is generated).

From this follows that in order to compute all possible resulting processes, `correct()` has to adjust a process and its entities in all possible ways resulting in an adapted process for every possibility. That means in the example described above, A1 is connected to A2, A3, XorJoin, and XorSplit resulting in four process copies.

Each of the resulting processes is checked for violated constraints. As soon as a violation is found, the constraint's `correct()` copies the process for each possible correction and applies the corrections in the individual copies.

By applying this procedure a directed graph consisting of process copies is created. The root of the graph corresponds to the initial process adapted by the BIO. The edges of the graph correspond to process adaptations by `correct()` of violated constraints.

In order to implement this procedure, the architecture of Fig. 2 has to be extended: `Correct()` has to create for each possible process adaptation a copy of the process, adapt it, and eventually return all copies. Returned copies are checked whether they violate constraints and if so, `correct()` of this constraint is called. This procedure is continued until all constraints are satisfied. Processes satisfying all constraints are saved and presented to the user which can decide which variation of the process to adopt after all solutions have been found. The algorithm for correcting a process is shown in Listing 1.

An instance of process (original) adapted by a BIO is passed to the method `controlCorrection()`. After creating a container for correct resulting processes (solutions) and a stack for storing the adapted but incorrect process copies (lifo), `checkAll()` is called on original returning the first violated constraint or `NULL` if no constraint is violated. In the latter case there is no need for further adaptations and the original can be sent back to the caller as solution. In case there are violations, original is pushed in the stack.

```
controlCorrection(Process original){
    Container solutions;
    Stack lifo;
    Constraint con = original.checkAll();
    if(con == null)
        solutions.add(original);
    else
        lifo.push(original);
    while(lifo.size()!=0){
        original = lifo.pop();
        Constraint vio = original.getViolation();
        Container adaptedProcesses = vio.correct();
        if(adaptedProcesses != null){
            for(Process p : adaptedProcesses){
                Constraint con = p.checkAll();
                if(con != null)
                    lifo.push(p);
                else
                    solutions.add(p);
            }
        }
    }
    return solutions;
}
```

Listing 1 Correction algorithm

Until there are instances of process in stack life, the next instance is taken and `correct()` of its violation is called. The returned adapted processes are checked. If there are violations in a process, it is pushed in the stack; otherwise the copy is added to solutions which is eventually returned to the caller.

2.4 Particular Design Decisions

In order to make the approach more efficient a number of particular design decisions have been made including control of loops and duplicates in (1) the graph and (2) priority management of constraints.

2.4.1 Loop and Duplicate Control

Life-locks in the graph and duplicated processes in solutions might occur. The former can occur when constraints loop. Constraints loop when a `correct()` adapts a process leading (not necessarily immediately) to a violation whose `correct()` causes a violation triggering the first `correct()` again. Duplicates in the resulting process solutions occur when two branches in the graph of processes join and the path leading from this spot is investigated more than once.

A control mechanism avoiding such situations has to find out whether an identical process exists to a newly created and adapted process in the graph. This can be expensive since processes have to be compared with all processes in the graph.

However, a loop or joining branches can also be detected when a violation occurs more than once on the same object (i.e., entity in case it is a local constraint) on the path from the root to the current process in the graph. This approach is less expensive since only violations have to be compared. However, for it to work, `correct()` must solve a violation entirely (i.e., the constraint is not violated after the adaptation any more) otherwise a loop would be detected that does not exist.

The latter approach is less expensive and `correct()` can be defined to satisfy the restriction to correct a violation entirely. For controlling loops and duplicates in solutions we therefore propose to check whether the same violation has already occurred in any ancestor of the process in the graph and if so, the path is no longer continued.

2.4.2 Priority Management of Constraints

Constraints can be very expensive to check. Consider the constraint that an artifact *art* has to be created by an activity *act* before it can be input of activities. For checking this constraint it has to be analyzed whether all activities having an input information-flow with *art* can be reached following the control-flows leading from *act*. All possible paths leading from *act* to end-events have to be investigated. It is desirable to check such expensive constraints as rarely as possible which can be accomplished by prioritizing constraints.

The algorithm described in Listing 1 stops at the first violation in the process, corrects it, and neglects possible further violations in the process. From this follows that violations can occur in more than one branch of the graph when there is more than one violation in a process since only one violation is corrected at a time and when the preceding correction resulted in more than one adapted process. In most cases this is not an issue since most *correct()* can be run without human interaction and require only minor computational resources. However, some require user input (e.g., provide resource for executing activity) and users might be asked to provide the same information more than once. It is the case when there is a violation requiring user input in the process but this violation is not corrected first.

A plausible work around is to cache user input. Another solution is to move constraints whose *correct()* requires user input as far up in the graph as possible (i.e., prioritize them), making them less likely to be executed more than once. However, this does not guarantee that a user is only asked once for a particular information. For example: Two violations requiring user input have occurred at the same time in a process. Only one of them can be corrected first. For the second the user might have to insert the information more than once since it will exist in all resulting process copies of the correction of the first violation.

However, since this occurs rarely, we propose to prioritize constraints where

- highest priority have constraints requiring user input for their correction,
- modest priority have constraints requiring no user input and with low or modest expensiveness for checking, and
- lowest priority have constraints requiring no user input for correction, but being expensive to check.

This enables checking expensive constraints as rarely as possible and minimizes multiple insertion of user input.

3 Evaluation

A prototype of a system for instantiating a reference process described above has been implemented for a particular Siemens business unit. The business unit uses a process comprising 23 types of entities including phases (composite activities consisting of a subprocess), milestones, and control-flows. The method manual comprises furthermore several subtypes of activities, resources, artifacts, splits, joins, events, information-flows, and associations of resources with activities (called resource connections). From the textual method manual, 135 constraints have been identified on types of entities of the meta-model and corresponding *correct()* methods have been defined. The reference process used for testing comprises about 3,000 instances of entities. The 135 constraints defined on types of entities result in about 14,000 constraints on instances of entities which have to be checked and, if necessary, corrected when violated during instantiation.

Fifteen BIOs defined by experts necessary for instantiation of the reference process of the business unit have been implemented. The operations are as elementary as possible in order to reduce complexity and avoid dependencies with other operations. Because of this simplicity, it might be necessary to execute more than one BIO to accomplish a complex adaptation step. For details refer to [15].

Siemens uses ARIS Toolset for defining their reference processes. In order to avoid limitations due to APIs provided by ARIS, export functionality has been implemented which allows export of Siemens reference processes from ARIS toolset in XPDL [21]. XPDL has been chosen since it is powerful enough for storing all data contained in Siemens processes and since it is a standard enabling transfer of process data to and from many process tools. Process data in XPDL is input for the instantiation and correction system described above which has been implemented as Java application with a graphical user interface for representation of processes and for user control. Adapted and corrected processes are graphically presented by the GUI and can be saved in XPDL.

Intensive testing proved feasibility of the approach. For example: When running the BIO “inserting a parallel/alternative path” in the same setting as described in Section 2.3, correction took about 2.5 s. This is remarkable since recovering from this scenario can be regarded as one of the most expensive ones.

With a complete system available, more testing is planned in real-world projects at Siemens AG.

4 Related Work

Instantiation of processes to project-specific needs has been subject to intensive research in recent years. However, in early software process approaches it was thought that a perfect process can be developed which fits all software developing organizations and all types of projects [8]. It was soon recognized that no such process exists [5, 17]. This leads to the introduction of reference processes as general guidelines which are adapted for specific project needs.

Early approaches to overcome this problem have been developed, for example, by Boehm and Belz [8] and Alexander and Davis [1]. The former used the Spiral Model to develop project-specific software processes. The latter described 20 criteria for selecting the best-suited process model for a project.

Many different adaptation approaches have been proposed since then and the need for adaptation of processes is recognized in industry which is shown by a high number of publications about tailoring approaches in practice, e.g., Bowers et al. [9] and Fitzgerald et al. [12].

Although much effort has been put into improving the adaption of software processes to project-specific needs, the approaches proposed so far still suffer from important restrictions and none has evolved into an industry-accepted standard.

An important reason is the variety of meta-models for processes used in practice. For instance, Yoon et al. [22] developed an approach for adapting processes

in the form of Activity-Artifact-Graphs. Since the process is composed of activities and artifacts, only the operations “addition” and “deletion” of activities and artifacts are supported as well as “split” and “merge” of activities. Another example is the V-Model [7], a process model developed for the German public sector. It offers a toolbox of process modules and execution strategies. The approach for developing a project-specific software process is to select required process modules and an execution strategy. Due to these dependencies on the meta-models, none of the existing approaches offers a complete and semiautomated method.

Because of the close relationship between Siemens software processes and business processes, adaptation approaches for the latter are also of interest. Approaches for processes and workflows of higher complexity are often restricted to only a subset of adaptation operations. For instance, Rosemann and van der Aalst [18] developed configurable EPCs (C-EPCs) enabling the customization of reference processes. However, the approach only allows activities to be switched on/off, the replacement of gateways, and the definition of dependencies of adaptation decisions.

Armbrust et al. [3] developed an approach for the management of process variants. A process is split up in stable and variant parts. The latter depend on project characteristics and are not allowed to be dependent on each other. The process is adapted by choosing one variant at the start of a project. Although the need for further adaptations during the execution of the process has been identified, no standardization or tool support is provided.

Allerbach et al. [2] developed a similar approach called Provop (Process Variants by Options). Processes are adapted by using the change operations *insert*, *delete*, *move*, and *modify attributes* which are grouped in Options. Options have to be predefined and are used to adapt processes, but they do not guarantee correctness.

In conclusion, none of the existing approaches offer a comprehensive, flexible, and semiautomated adaption of processes as required for the diversity of processes and software development encountered in large enterprises.

5 Conclusions

Siemens is currently undertaking research efforts to further improve its software process-related activities. Part of these efforts is the development of a system that supports project managers in instantiation of reference processes. The system aims not only to execute decisions but to restore correctness of the resulting process when violated by the execution of the decision. Since the implementation of such a system is organization-specific and depends on the permitted elements and constructs in the process, a flexible architecture has been developed and described. Its feasibility has been verified by the implementation of a prototype. Future work will include enhancement of the prototype and its evaluation in software development projects at Siemens AG.

References

1. Alexander, L. and Davis, A. (1991) Criteria for Selecting Software Process Models, in: *Proceedings of the 15th Annual International Computer Software and Applications Conference*, pp. 521–528.
2. Allerbach, A.; Bauer, T.; and Reichert, M. (2008) Managing Process Variants in the Process Life Cycle, in: *Proceedings of the 10th International Conference on Enterprise Information Systems*, pp. 154–161.
3. Armbrust, O.; Katahira, M.; Miyamoto, Y.; Münch, J.; Nakao, H.; and Ocampo, A. (2008) Scoping Software Process Models – Initial Concepts and Experience from Defining Space Standards, in: *ICSP*, pp. 160–172.
4. Bandinelli, S. and Fuggetta, A. (1993) Computational Reflection in Software Process Modeling: The SLANG Approach, in: *ICSE*, pp. 144–154.
5. Basili, V. and Rombach, H. (1991) Support for Comprehensive Reuse, *Software Engineering Journal* 6(5), 303–316.
6. Becker, U.; Hamann, D.; and Verlage, M. (1997) Descriptive Modeling of Software Processes, in: *Proceedings of the 3rd Conference on Software Process Improvement (SPI '97)*.
7. BMI (2004) The new V-Modell XT – Development Standard for IT Systems of the Federal Republic of Germany, URL: <http://www.v-modell-xt.de> (accessed 05.05.2009).
8. Boehm, B. and Belz, F. (1990) Experiences with the Spiral Model as a Process Model Generator, in: *Proceedings of the 5th International Software Process Workshop 'Experience with Software Process Models'*, pp. 43–45.
9. Bowers, J.; May, J.; Melander, E.; and Baarman, M. (2002) Tailoring XP for Large System Mission Critical Software Development, in: D. Wells and L. Williams, ed., *Extreme Programming and Agile Methods – XP/Agile Universe 2002*, 2nd XP Universe Conference Chicago, pp. 100–111.
10. Brinkkemper, S. (1996) Method Engineering: Engineering of Information Systems Development Methods and Tools, *Information & Software Technology* 38(4), 275–280.
11. Feiler, P. H. and Humphrey, W. S. (1993) Software Process Development and Enactment: Concepts and Definitions, in: *ICSP*, pp. 28–40.
12. Fitzgerald, B.; Russo, N.; and O’Kane, T. (2000) An Empirical Study of Systems Development Method Tailoring in Practice, in: *Proceedings of the 8th European Conference on Information Systems*, pp. 187–194.
13. IBM (2008) Rational Method Composer, URL: <http://www-01.ibm.com/software/awdtools/rmc/> (accessed 05.05.2009).
14. Kabbaj, M.; Lbath, R.; and Coulette, B. (2008) A Deviation Management System for Handling Software Process Enactment Evolution, in: *ICSP*, pp. 186–197.
15. Killisperger, P.; Peters, G.; Stumptner, M.; and Stückl, T. (2009), *Information Systems Development: Towards a Service Provision Society*, Springer (2009), Chapter Instantiation of Software Processes, An Industry Approach, pp. 589–597.
16. Killisperger, P.; Stumptner, M.; Peters, G.; and Stückl, T. (2008) Challenges in Software Design in Large Corporations – A Case Study at Siemens AG, in: *Proceedings of the 10th International Conference on Enterprise Information Systems*, pp. 123–128.
17. Osterweil, L. J. (1987) Software Processes Are Software Too, in: *ICSE*, pp. 2–13.
18. Rosemann, M. and van der Aalst, W. (2007) A Configurable Reference Modelling Language, *Information Systems* 32(1), 1–23.
19. Scheer, A. (2000) *ARIS- Business Process Modeling*, Springer, Berlin.
20. Schmelzer, H. and Sesselmann, W. (2004) *Geschäftsprozessmanagement in der Praxis: Produktivität steigern – Wert erhöhen – Kunden zufrieden stellen*, Hanser Verlag, Muenchen.
21. WfMC (2008). WfMC-TC-1025-Oct-10-08A (Final XPDL 2.1 Specification). URL: <http://www.wfmc.org> (accessed: 28.04.2009).
22. Yoon, I.; Min, S.; and Bae, D. (2001) Tailoring and Verifying Software Process, in: *APSEC*, pp. 202–209.

Adopting Quality Assurance Technology in Customer–Vendor Relationships: A Case Study of How Interorganizational Relationships Influence the Process

Lise Tordrup Heeager and Gitte Tjørnehøj

Abstract Quality assurance technology is a formal control mechanism aiming at increasing the quality of the product exchanged between vendors and customers. Studies of the adoption of this technology in the field of system development rarely focus on the role of the relationship between the customer and vendor in the process. We have studied how the process of adopting quality assurance technology by a small Danish IT vendor developing pharmacy software for a customer in the public sector was influenced by the relationship with the customer. The case study showed that the adoption process was shaped to a high degree by the relationship and vice versa. The prior high level of trust and mutual knowledge helped the parties negotiate mutually feasible solutions throughout the adoption process. We thus advise enhancing trust-building processes to strengthen the relationships and to balance formal control and social control to increase the likelihood of a successful outcome of the adoption of quality assurance technology in a customer–vendor relationship.

Keywords Quality assurance · CMMI · ISO · GAMP · Interorganizational relationships · Trust · Control

1 Introduction

Quality assurance has its roots in Total Quality Management (TQM) [1] and is a formal behavioural control mechanism [2, 3] aiming at increasing the quality of an industrial or tailor-made product exchanged between vendors and customers. In the field of IT systems development, we find two main approaches to quality assurance and a row of combinations and adaptations of these. CMM(I) [4, 5] is the main approach to the improvement of software processes, while the ISO 9000

L.T. Heeager (✉)
Aalborg University, Aalborg, Danmark
e-mail: liseh@cs.aau.dk

series of standards [6] and QFD [7] are the main standards for quality improvement. BOOTSTRAP [8] is an example of an approach that builds on both the main principles.

Many IT vendors strive to become certified as compliant with the different standards and norms as they hope it will secure them a beneficial market position, while others just aim at improving their work processes and raising the quality of their products. Often, these companies initiate this effort as part of their long-term business strategy and not in response to the requirements of their customers.

The adoption of quality assurance can be studied either with a focus on adoption in the vendor organization or on the customer requirements and validation work or focussing on the role of the relationship between the two in the adoption process. Interestingly, CMM has norms both for vendors improving software processes and for customers improving their acquisitions process, but not for the mutual process [9]. Only little research has focussed on the relationship between customers and IT vendors when adopting quality assurance technology.

We have therefore investigated how the process of adopting quality assurance standards is influenced by the relationship between a customer and an IT vendor. We achieve this through a case study of the adoption of the GAMP standard [10] by a small Danish IT vendor (PharmSoft) developing pharmacy software for a customer in the public sector. This customer had geographically distributed user sites, each with separate management, but the cooperation with PharmSoft was organized and managed through a central department. The project management of PharmSoft worked closely with this department describing system requirements, while a cross-organizational steering committee prioritized the development resources. We study this case as the adoption of quality assurance technology in an interorganizational relationship between the vendor and the customer [11].

The chapter is structured as follows. First, we introduce the GAMP standard and the validation process for computer systems in Section 2. Then, we present the theory of interorganizational relationships, focussing on the concepts of trust and control [2, 3] and further attributes and processes [11] in Section 3. After describing our research approach (Section 4), we present the analysis of the adoption of quality assurance in PharmSoft (Section 5) followed by an analysis of the customer–vendor relationship of PharmSoft in Section 6. The interorganizational relationship and its influence on the adoption process are discussed in Section 7 and the conclusion is found in Section 8.

2 Quality Assurance in the Pharmaceutical Sector and the GAMP Standard

In Denmark, suppliers within the pharmaceutical sector are required to validate the quality of their delivery to the public sector. This includes IT systems dealing with the production and handling of medicine if they are part of such supplier

systems. The GAMP4 Guide for Validation of Automated Systems launched in December 2001 by ISPE (International Society for Pharmaceutical Engineering) is widely used in Europe and was chosen as the standard by the customer in this case. GAMP4 is part of the GxP family of standards (Good Practice quality guidelines and regulations), which is used by the pharmaceutical and food industries and includes Good Manufacturing Practice (GMP), Good Laboratory Practice (GLP), Good Documentation Practice (GDP) and Good Clinical Practice (GCP) [10].

The central activity of GAMP is the validation of the automated system to ensure correct operation by a documented proof. To the extent that the automated system is an IT system, the system development process and its products should be validated too.

In comparison with the ISO 9001 standard, GAMP4 has stricter requirements for traceability from the customer requirements to the implemented system changes, for more detailed specification documentation and for standard operating procedures [12].

3 Interorganizational Relationships

At the beginning of this study, we focussed on the adoption process of the vendor but experienced that this picture did not reveal the whole situation. Thus, we have studied the adoption process of quality assurance as being joint between the customer and the vendor and thus being acted out within their relationship. We base our analysis on the theory of interorganizational relationships [2, 3] to understand this adoption process from both sides and its outcome.

The nature of an interorganizational relationship in general has a substantial influence on how successful the outcome of the partnership/cooperation is [2, 11, 13, 14] and how high the level of confidence in a partner cooperation is [2]. Both trust and control can contribute to building confidence in partner cooperation in interorganizational relationships. “A higher trust level does not automatically dictate a lowering of the control level and vice versa” [2].

Quality assurance technology is a formal behavioural control mechanism. Control mechanisms are “organizational arrangements designed to determine and influence what organization members will do” [2]. Formal control is an external measure-based control that “emphasizes the establishment and utilization of formal rules, procedures and policies to monitor and reward desirable performance” [3] in opposition to social control (or informal control or internal value-based control), which relies on shared norms, values, culture and internalization of goals as means to reach the desired outcome.

Trust plays an important role in achieving a successful relationship [11, 13–17]. Trust is the “positive expectations about another’s motives with respect to oneself in situations entailing risk” [2]. Goodwill trust is trust in one’s good faith, good intentions and integrity, whereas competence trust is trust in one’s ability

to do appropriate things, not trust in the other party's intention to do so. Having goodwill trust in a relationship, the partner will be less concerned with problems of cooperation [3]. It is noticeable that trust is hard to build, but easy to violate [15].

Both formal and social control can be successfully implemented in an interorganizational relationship, but may have different impacts on the level of trust. Formal control mechanisms can be a hindrance to a high level of trust, through their constraining of people's autonomy, whereas social control as a by-product can be trust-building as it often takes the form of socializing and interaction. There may be a contradistinction between trust and formal control mechanisms such as quality assurance when it comes to building confidence in a relationship.

Complete trust in a partner could point to no or a very low need for control mechanisms, but exclusive reliance on trust in relationships introduces an increased risk of failure, as structures or potential problems will be disregarded. Relying on excessive structural control can, however, hurt performance. Most projects therefore need a balance between trust and control [16].

To understand the influence of the nature of the relationship on the adoption process, we also include more parameters in our study of the relationship. We adopt the attributes and processes of Goles and Chin [11]. Through a thorough literature review of previous research, they constructed a list of factors playing a significant role in relationships and grouped these into attributes and processes. The five attributes that contribute to the functionality and harmony of the relationship are commitment, consensus, cultural compatibility, flexibility and interdependence and the five processes that develop the attributes are communication, conflict resolution, coordination, cooperation and integration [11]. Our literature study of papers on interorganizational relationships confirmed the conclusion that these attributes and processes are those most significant for a successful relationship [2, 3, 13–17].

Figure 1 is a framework illustrating the process of adopting quality assurance standards in an interorganizational relationship and the influencing factors. The adoption process takes place and is influenced by the attributes and processes of the relationship between the customer and the vendor. However, the context in which this relationship is acted out will also influence the adoption process. As time passes

Context

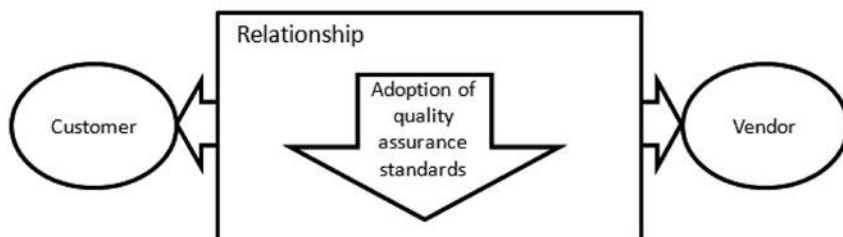


Fig. 1 Adopting quality assurance standards in an interorganizational relationship

and the adoption process develops, the relative importance of the influencing factors, e.g. the attributes and processes [11], changes.

4 The Research Approach

The case study was organized as an interpretive single case study [18] and ran from September 2007 until June 2008, just before and after the first external audit of the vendor (see Fig. 2). Our research interest was in understanding the process of adopting quality assurance.

The data were collected and analysed in two phases. This gave us a rich history on which to base our data collection and analysis. In the first phase, we studied the adoption of the GAMP standard from the view of the vendor and focussed on the internal processes. The interviews included employees from PharmSoft: the project manager and three developers. They were semi-structured and organized as diagnostic interviews [19]. The purpose was to gain an initial understanding of the adoption process of the vendor and to identify the employees’ opinions and views on the situation. Furthermore, one researcher made observations of the vendor for 4 months, building personal knowledge of the case. This phase showed us that studying this process only from the vendor’s side would not reveal a full picture of the situation. Thus, we chose to involve the customer and focus on the interorganizational relationship.

In the second phase, we shifted the level of focus and studied the adoption process from an interorganizational perspective. The phase included five qualitative interviews with the project manager from PharmSoft, a developer from PharmSoft, two users of the system and a customer representative. These interviews were semi-structured and explicitly seeking the interviewee’s personal view of the situation.

During the two phases, all the interviews were recorded and transcribed and documents relevant to the adoption process from both the vendor and the mutual project were collected.

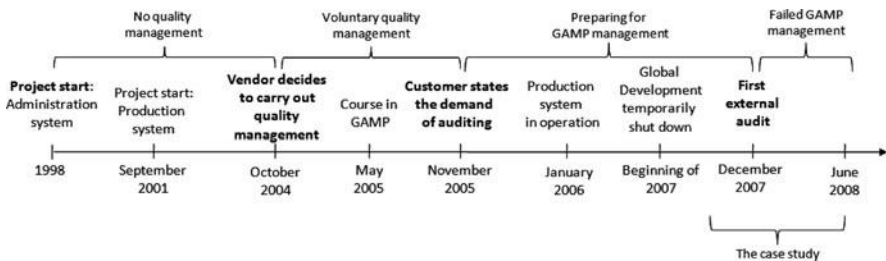


Fig. 2 Timeline

We analysed the data iteratively through two analyses both in between the phases and after the second phase. First, we compiled a historical map (see Fig. 2) of important events in the adoption process by working through the data and discussing them with the project manager of the IT vendor. We identified four phases: no quality management, voluntary quality management, preparing for GAMP management and failed GAMP management. Based on this, we wrote up the case story as seen in Section 5. Then, we analysed the customer–vendor relationship, according to the attributes and processes of Goles and Chin [11], with special emphasis on trust and control [2, 3]. We investigated each interviewee’s perception of the relationship to clarify whether there were different perceptions between the customer and vendor and to clarify any internal differences. The findings were written up and presented to the interviewees. To our surprise, the perception of the relationship was shared overall between the parties; thus, we have reported the results in Section 6 as one description, though pointing out the differences.

The interpretation of this case emerged during the rather elaborate analysis process and mainly builds on the concepts of trust and control [3].

5 Adoption of Quality Assurance at PharmSoft

The adoption process had the form of an iterative long-term and somewhat experimental adoption process resulting from ongoing negotiations between the two parties of the customer–vendor relationship.

The system development process was initiated in 1998. The first subsystems developed did not require validation of the software. In September 2001, the planning of a production subsystem was initiated, and during the next 2 years, the parties realized that this system would be the target of validation and they discussed the subject initially and briefly. However, no concrete requirements for the validation process were made and no quality assurance standards were decided on, since the customer organization was in doubt regarding how to handle the legal requirements.

In light of this, PharmSoft decided in October 2004 to initiate a quality assurance effort on its own. Inspired by traditional quality assurance practice in the IT industry, the employees described their existing working practices in formal procedures to document their initial quality level. The main focus of this work was system documentation, document management, requirements and change management. They organized the adoption process with the person responsible for the quality assurance, writing up initial procedures that were reviewed, discussed and even changed in workshops with the employees. The customer organization was of course informed about this work and the results.

The same autumn, at a steering committee meeting, a powerful manager of the customer organization presented and advocated the GAMP4 standard as an appropriate choice for organizing the quality assurance effort. This sparked a clarification in the customer organization on how to meet the public requirements and after some time the choice of the GAMP standard was approved. Both the customer and

PharmSoft made an effort to learn and understand the GAMP standard, including a joint course on GAMP in May 2005. Eventually, only 2 months before the production system was implemented in November 2005, the customer organization made an official demand for a formal audit according to the GAMP standard. Neither the customer organization nor PharmSoft, however, knew what that actually meant, since both lacked experience in interpreting a quality standard and design or even implementing procedures according to it, even though both had responsibilities in the validation work according to the standard.

PharmSoft prepared well for the audit, shaping up the existing quality assurance system and striving to implement procedures to meet the rest of the standard requirements to do as well as possible. The customer organization, on the other hand, did not expect PharmSoft to pass the audit at the first attempt. They had called for it as part of the clarification of how to implement GAMP in the project.

PharmSoft failed the audit by external and independent auditors, with remarks mostly on the quality assurance organization, test and test documentation, document management requiring much more printed and signed documentation (contrary to the existing online documentation) and requirement specification. Both test and requirement specifications are shared working practices between the customer organization and PharmSoft. An example is how the customer and PharmSoft collaborated and negotiated the design of the new test procedure: the test practice until December 2007 had been developed over years by the parties until they agreed on having reached a sufficiently high level of quality of the system implemented in each delivery. However, since GAMP had stricter requirements for documentation and for the separation of validation and software development, a new test practice was launched in spring 2008. The first test conducted thereafter did not work as expected. Too many flaws were found too late in the test period. As a result, the customer and PharmSoft decided to integrate the new procedure with the old test practice into yet another new test procedure for the system.

After the audit, both parties decided to buy in external experts on quality assurance to help them reach an appropriate design and implementation of an appropriate quality assurance system complying with GAMP4, but highly adapted to existing practice in the project. As of November 2008, PharmSoft had not yet passed the certification, but the adoption process was still progressing to both parties' satisfaction.

6 Analysis of the Interorganizational Relationship of PharmSoft

The relationship is analysed according to the five attributes and processes (Tables 1 and 2). The customer and PharmSoft describe the level of trust between them as high. Their communication and cooperation is characterized by willingness, honesty and flexibility. By these means, they resolve conflicts peacefully, preventing problems from growing too big. Through the years, they have built a high mutual knowledge and come to consensus in many fields. The customer is

Table 1 Summary of the findings from the attributes analysis of the case

Attributes	
Interdependence	PharmSoft is highly dependent on the contract with the customer organization, since it is its main customer. However, since the system has become very complex and difficult to maintain over the years, both the project manager and the developers from PharmSoft perceive this dependency to be bilateral. The less trusting parts of the customer organization have, however, been giving thought to looking for a substitute vendor to keep down the expenses and raise the quality of the system
Flexibility	The main responsible person from the customer and the system users perceive PharmSoft as flexible, when circumstances change. Limited resources (the employees) combined with the complexities of the tasks, however, restrict this flexibility. The customer representative accepts this as reasonable
Cultural compatibility	The cultural compatibility is high since both PharmSoft and the customer are experienced actors on the Danish IT market and thus overall share this business culture, knowing the basic legal requirements and formal and informal rules of practice in the market
Commitment	PharmSoft is very committed to the relationship, according to the customer representatives. It does all it can to please the users. The representative emphasized the willingness of PharmSoft to discuss problems and find solutions On the other hand, commitment from the customer towards the relationship is diverse. The parts of the organization working closely with PharmSoft are highly committed but other parts show some resistance, as mentioned under the other attributes
Consensus	The relationship is characterized by a remarkable consensus between the parties The customer and PharmSoft express an overall unity built through a mutual understanding and close collaboration over time. Their frequent communication and dialogues help build this consensus Within the customer organization, the conflict of interest is evident. The central department and close vendor contact prioritize a swift adoption of GAMP, while the user sites prioritize the development of new and added functionality. In between, PharmSoft finds it difficult to know whom to serve first

Table 2 Summary of the findings from the attribute-building-process analysis of the case

Processes	
Cooperation	PharmSoft and the customer describe their cooperation as close and well functioning. The vendor often successfully pleases the users and the vendor feels understood by the customer
Communication	The communication between PharmSoft and the customer representative is frequent. They communicate to manage the development, to confirm requirement specifications and whenever an issue has to be discussed. Both parties perceive their communication as honest

Table 2 (continued)

Processes	
	PharmSoft communicates with the system users at the steering committee once a month and when local system adaptations or add-ons need to be negotiated. This latter communication is either by phone or email. Both PharmSoft and the customer prefer communicating by phone as they find that misunderstandings occur more often in written form
Conflict resolution	Both the customer and PharmSoft said that they have no conflicts, only small disagreements that are solved through communication
Coordination	Coordination was not an important process of this case
Integration	Integration was not an important process of this case
Creating mutual knowledge	The mutual knowledge is highest between PharmSoft and the customer representative and between PharmSoft and the system users. This knowledge has been built through time, mainly at the monthly steering committee meetings that involve a stable group of people. Mutual understanding was mentioned by both PharmSoft and the customer as being important for a successful relationship

furthermore satisfied overall with the system delivered by PharmSoft. We will therefore characterize this as a trusting relationship. However, some parts of the customer organization are less satisfied with the whole arrangement, which puts the relationship under some kind of pressure.

7 Discussion

The relationship between the customer and PharmSoft is characterized overall by both mutual goodwill trust and competence trust as both parties expressed high trust in the other party’s good intention, commitment to the cooperation and ability to cope with the tasks. The customer expressed the belief that PharmSoft does everything possible to please them, and the further belief that PharmSoft will achieve the validation within an acceptable time limit. PharmSoft accepted the quality assurance management and appreciated the fact that the customer negotiated the requirements of the standard.

The high level of trust has been built through years of successful, close cooperation involving a lot of direct communication between a stable group of people including the project management of PharmSoft and the customer representative and several of the system users.

The relationship between the customer and PharmSoft was mostly based on social control and high mutual trust already before starting to develop the production system. The choice of quality assurance standard and thereby the official beginning of quality assurance was diminished due to the trusting relationship. Neither the customer nor PharmSoft experienced a need for further control. The need for validation arose solely due to the legal requirement.

The voluntary quality management was triggered when PharmSoft realized that the subsystem would be subject to the legal requirement for validation. To meet a demand from its customer, it initiated a quality management effort focussing on immediate beneficial improvements in its work practice. This worked as a sign of flexibility and willingness to cooperate from PharmSoft.

The next months showed that the adoption of quality assurance technology was perceived as a mutual task by the parties and that they cooperated in order to ease the process. We find evidence that the driver of this was the trusting relationship as shown by the fact that PharmSoft decided to initiate the quality management on its own, in the joint course on GAMP and in the way it negotiated the requirements for the quality management.

The process that led to the decision on GAMP was sparked by a manager at the customer, who as an outsider of the trusting relationship pushed the model and the need for orderly validation on a steering committee meeting. We do not know how and when the decision was made, but it ran seamlessly into the third phase: preparing for GAMP quality management. The fact that it took an outsider to make the official decision on quality assurance shows that trust can diminish the felt need for the implementation of a formal control mechanism in a well-functioning relationship. Subsequently, a period of preparing for the upcoming audit began. Even though PharmSoft made an effort to adopt the quality management of GAMP, it was well known prior to the audit on both sides that it was likely to fail.

As expected, PharmSoft failed the audit and, as the problems now were explicitly known, the customer and PharmSoft were forced to deal with the situation. However, they managed to overcome this situation by cooperating and found a mutual, satisfactory solution, e.g. this was shown in the way they handled the negotiation of the test procedures (see Section 5).

The case of PharmSoft showed that an interorganizational relationship may influence the process of quality assurance adoption in several ways. The level of trust can impact on the felt need for more formal control to an extent where both of the participants are ready to rely solely on trust and, as shown in this case, trust between parties can diminish the formal decision of quality assurance adoption. However, trust between parties in an interorganizational relationship is very helpful when adopting quality assurance. Trust makes the parties perceive the adoption process as a mutual task and cooperate to smooth the adoption process for both parties, reducing the number of problems owing to conflicts and helping in the process of resolving the conflicts that do arise. Due to few conflicts, the quality and time of delivery of the outcome will not be pressured. If the adoption process is successful, this mutual process will enhance the trust level even further.

8 Conclusion

In this chapter, we investigate how the process of adopting quality assurance standards is influenced by the relationship between a customer and an IT vendor. This

was achieved through an interpretive case study of an IT vendor striving to adopt GAMP in a relationship with its customer.

From this case, we learn that a successful trusting relationship between an IT vendor and a customer can be threatened by the adoption of the formal control mechanism quality assurance. However, we also found that the trusting and close relationship helped the parties to overcome this threat and integrate the quality assurance technology into their relationship.

Our research shows that not only the parties and their internal capacities matter in the adoption of quality assurance technology, but that the relationship between them and the surrounding context also influence the process itself and the outcome. We have presented a framework illustrating the adoption of quality assurance in an interorganizational relationship based on the theory of interorganizational relationships [11], which we found useful in the analysis of the case. We suggest that further studies could adopt this broad scope for understanding the adoption of quality assurance technology.

Even though it was not our purpose, we found indications in our study that adopting quality assurance to a large extent does influence the customer–vendor relationship and that especially relationships that are based on trust can be harmed. This may be a somewhat neglected consequence of adopting quality assurance technology that deserves more attention in future research.

References

1. Deming, W. E. (1982). *Out of the Crisis*. Cambridge, MA: Productivity Press.
2. Das, T. K., and Teng, B.-S. (1998). Between trust and control: Developing confidence in partner cooperation in alliances. *Academy of Management* **23**, 491–512.
3. Das, T. K., and Teng, B.-S. (2001). Trust, control, and risk in strategic alliances: An integrated framework. *Organization Studies* **22**, 251–283.
4. CMMI Product Team. (2001). *Capability Maturity Model Integration*. Pittsburgh, PA: Carnegie Mellon Software Engineering Institute.
5. Paulk, M. C., Chrissis, M. B., and Weber, C. (1993). *Capability Maturity Model for Software Version 1.1*. Pittsburgh, PA: Software Engineering Institute.
6. Hoyle, D. (2005). *ISO 9000 Quality Systems Handbook*. Boston: Elsevier Science and Technology.
7. Akao, Y., and Mazur, G. H. (2003). The leading edge in QFD: Past, present and future. *International Journal of Quality & Reliability Management* **20**, 20–35.
8. Kuvaja, P., and Bicego, A. (1994). BOOTSTRAP – a European assessment methodology. *Software Quality Journal* **3**, 117–127.
9. Bjerknes, G., and Mathiassen, L. (2000). Improving the Customer-Supplier Relation in IT Development. *Proceedings of the 33rd Hawaii International Conference on System Sciences*, 1–10.
10. ISPE. (2008, September 29). *International Society for Pharmaceutical Engineering*. Retrieved from <http://www.ispe.org>.
11. Goles, T., and Chin, W. W. (2005). Information systems outsourcing relationship factors: Detailed conceptualization and initial evidence. *The DATABASE for Advances in Information Systems* **36**, 47–67.
12. Wright, G. (2003). *Achieving ISO 9001 Certification for an XP Company*. Berlin, Heidelberg: Springer, 43–50.

13. Holmström, H., Conchúir, E. Ó., Ågerfalk, P. J., and Fitzgerald, B. (2006). The Irish Bridge: A case study of the dual role in offshore sourcing relationships. *Twenty-Seventh International Conference on Information Systems*, 513–526.
14. Kern, T. (1997). The gestalt of an information technology outsourcing relationship: An exploratory analysis. *International Conference on Information Systems*, 37–58.
15. Brunard, V., and Kleiner, B. H. (1994). Developing trustful and co-operative relationships. *Leadership & Organization Development Journal* **15**, 3–5.
16. Sabherwal, R. (1999). The role of trust in outsourced IS development. *Communications of the ACM* **42**, 80–86.
17. Zaheer, A., McEvily, B., and Perrone, V. (1998). Does trust matter? Exploring the effects of interorganizational and interpersonal trust on performance. *Organization Science* **9**, 141–159.
18. Walsham, G. (1995). Interpretive case studies in IS research: Nature and method. *Operational Research Society Ltd*, 74–81.
19. Iversen, J., Nielsen, P. A., and Nørbjerg, J. (1999). Problem diagnosis software process improvement. *The DATABASE for Advances in Information Systems* **30**, 66–81.

A Framework for Decomposition and Analysis of Agile Methodologies During Their Adaptation

Gytenis Mikulenas and Kestutis Kapocius

Abstract In recent years there has been a steady increase of interest in Agile software development methodologies and techniques, which are often positioned as proven alternatives to the traditional plan-driven approaches. However, although there is no shortage of Agile methodologies to choose from, the formal methods for actually choosing or adapting the right one are lacking. The aim of the presented research was to define the formal way of preparing Agile methodologies for adaptation and creating an adaptation process framework. We argue that Agile methodologies can be successfully broken down into individual parts that can be specified on three different levels and later analyzed with regard to problem/concern areas. Results of such decomposition can form the foundation for the decisions on the adaptation of the specific Agile methodology. A case study is included in this chapter to further clarify the proposed approach.

Keywords Agile software development · Method engineering · Crystal Clear

1 Introduction

Although only 8 years have passed since the first publishing of the Agile Manifesto, the concept of Agile development has gained strong positions within the field of Information Systems Development (ISD). Such approaches as Extreme Programming, Scrum, DSDM, Crystal, FDD, ASD, OpenUP, Agile modeling, Iconix, Lean software development, and Pragmatic Programming are now being positioned as proven alternatives to the more traditional plan-driven approaches. However, although there is no shortage of Agile methodologies to choose from; the formal methods for actually choosing or adapting the right one are lacking. Majority of the researchers in this field concentrate on presenting success stories or lessons

G. Mikulenas (✉)

Department of Information Systems, Kaunas University of Technology, Kaunas, Lithuania
e-mail: gytenis.mikulenas@ktu.lt

learned by organizations that have adopted Agile methodologies for specific projects [18, 21, 26, 27]. Others propose adopting individual Agile practices only [6, 24]. Finally, there is a group of considerably more practical proposals [10, 11, 12, 14] that, as is described in this chapter, were reflected in our work. Therefore, the aim of the presented research was to define the formal way of preparing Agile methodologies for adaptation and creating an adaptation process framework or a guide. But at the same time it must be easy to learn and adopt solutions [28, 35].

It is important to stress that practitioners are rarely faced with the need to adapt entire Agile methodologies. Companies usually have their own know-how and do not want to rebuild processes from scratch. Instead, the common aim is to improve the adopted processes by introducing some specific parts of certain Agile methodologies. However, such methodologies are often presented by their authors as monolithic solutions. In this chapter, we try to show that these methodologies can still be broken down into individual parts that can be specified on three different levels and later analyzed with regard to problem/concern areas, thus presenting decision makers with a clear view on what could be adapted and why.

The following two sections of the chapter are devoted to our observations regarding two major issues of Agile adaptation: agility requirements and decomposition of Agile methodologies. Our proposed framework of the adaptation process is presented in Section 4, which is followed by the description of the case study and concluding remarks.

2 Environmental Agility Requirements

An important issue facing both researchers and practitioners is pinpointing the environments where Agile methods are most likely to yield good results and defining the characteristics of such environments. In other words, there is a need to know what the agility requirements are for the project environment.

First of all, papers on specific Agile methodologies typically include direct or indirect guidelines on what to take into consideration when choosing the methodology for a particular project [4, 5, 8, 14, 16, 17, 34]. The general requirements, however, were already defined in a widely accepted Agile Manifesto, that declared 16 key principles, including “Individuals and interactions over processes and tools,” “Working software over comprehensive documentation,” “Customer collaboration over contract negotiation,” or “Responding to change over following a plan” [3].

Another way of exposing aforementioned requirements is by juxtaposing Agile approaches with the plan-driven ISD methods. In such comparisons, Agile methodologies are usually presented as innovative and modern solutions, while the differences are emphasized by defining the requirements and conditions for the application of both Agile and plan-driven ISD methods [10, 11, 12, 20]. The problem is that in such studies requirements are often closely intertwined with Agile practices and techniques. For example, refactoring, onsite customer [8] or pair-programming, test-first, and daily stand-up meeting [20] can be interpreted as separate Agile practices.

When speaking about the requirements, one can consider such factors as trust in the developers, morale, and company's position on accepting changes in requirements. If these factors are missing, it may be difficult to adopt Agile. On the other hand, there are also specific criteria, such as the size of the development team or criticality of the developed system. Such criteria can be treated as environmental requirements. In any case, the adaptation of Agile methodologies starts with the preparation of the project environment. Obviously, it is difficult to expect that the real-life environment will meet the Agile requirements 100%. Therefore, it has to be noted that any Agile environment requirements should be considered as something defining ideal conditions, because in practice specific Agile methods can be successfully applied beyond their defined limits. However, only by knowing the requirements for such medium (ideal conditions) can we look for the so-called "sweet spots" where application of the Agile methodologies yields best results.

So, it can be safely said that the possible types of elements of the Agile methodologies (discussed in the following section of this chapter) and requirements or preferred characteristics of the project environment are two different things. We propose to define such requirements as being one of the following:

- virtues, philosophy, and views of the project team members that are required in order to successfully apply Agile methodologies and
- quantitative constraints defining ideal project parameters that guarantee optimal performance of the selected Agile methodology in a given project.

Based on these assumptions, we have collected a set of requirements for the Agile environments that is presented in Table 1. The ideal environment for Agile adaptation should meet all of those requirements.

As can be seen from the presented list of requirements, moving toward the development of software in the Agile manner means not only adapting and using Agile practices but also changing the way of thinking, the values, and philosophy. Literature on specific Agile methodologies usually includes more detailed descriptions of certain requirements, so, before moving on with methodology adaptation, the requirements should be reviewed.

3 Decomposition of ISD Methods

It is a well-known fact that ISD methods are not used as they ought to be in actual software development projects [9]. Done in an ad hoc manner, this leads to a large number of variations even on one method within a single organization [29].

Motivated by the prevalent belief that no one methodology fits all situations, Method Engineering (ME) was first introduced as a discipline to study engineering techniques for constructing, assessing, evaluating, and managing methods for developing Information Systems Development Method (ISDM) [25]. As time went by, these ideas evolved into a subdiscipline – Situational Method Engineering

Table 1 Requirements of the Agile environment

Source	Requirement	Value in ideal conditions
[3, 10, 11]	Primary goal	Working software
[8, 20, 34]	Software requirements	Changing, unclear, ambiguous
[11, 16]	Criticality	Preferable low or medium safety-critical products
[16]	Software type	Information systems (object-oriented technologies)
[11, 16, 20]	Size	Well matched to small products and teams; reliance on tacit knowledge limits scalability
[11]	Customer	Dedicated, colocated collaborative, representative, authorized, committed, and knowledgeable
[16, 34]	Personnel	Motivated and skilled individuals, self-organizing teams
[11]	Dynamism	Rapid value, responding to change, speed, simple design, and continuous refactoring
[3, 11, 16]	Culture	Thrive in a culture where people feel comfortable and empowered by having many degrees of freedom; balancing between discipline and chaos
[10, 16]	Technical	Technical environment with acceptable efforts on refactoring, appropriate tools for automated tests, configuration management, frequent integration
[4, 5]	Modeling	Do no more than the absolute minimum to suffice
[11, 16]	Communication	Tacit interpersonal knowledge, face-to-face communication
[3]	Company policy	Customer collaboration over contract negotiation

(SME). Prevalent research viewpoint here was the support of the creation of the meta-methods and integration of ISDM parts that together form a new situational method that is adapted to match given organizational settings or specific development projects [22]. The process guide included with most SME approaches consists of two major parts. The first part consists of common steps such as breaking ISD method into small elements, describing their attributes and relations with other elements, and putting them into a method element repository. The second part describes the process of selecting these elements from a repository for the assembly.

It must be noted though that there is still no agreement between SME researchers regarding what should be considered a method element. There are various terms used to define this concept, including “process fragment,” “product fragment,” “method chunk,” “body,” “interface,” or “element” [29]. Speaking of Agile methodologies, the situation is similar. Here, we can find such notions as “property,” “strategy,” “technique,” “process,” “cycle,” “work product,” “role” [14], “responsibility,” “practice,” “artifact” [1, 2], “actor,” “team,” “lifecycle,” “build,” “milestone,” “phase,” and “task” [31]. However, although the debate on the exact definition of a method element is still on, we can safely distinguish three core meta-classes: producers, work units, and work products. All three classes can be found in the Open Process Framework (OPF) [19], Software Process Engineering Model (SPEM) [30], and ISO/IEC 24744 [23] – the most used standards in the research area of SME.

Finally, it must be noted that despite being aimed at solving software development problems, SME solutions are often very difficult to adapt in real-world settings and are therefore criticized by the practitioners. Complexity of the process is one of the problems we tried to address in the proposal presented below.

4 Proposed Process of Agile Method Decomposition and Analysis During Its Adaptation

Based on the analysis results, a formal framework that describes how Agile methodologies could be prepared for adaptation (steps 1–4 in the following list) and supplies a basic adaptation process guide (step 5) was developed. The proposed process consists of the following steps (note that relevant examples can be found in Section 5 of this chapter).

4.1 Eliciting Environmental Requirements

During this step, Agile requirements presented in Table 1 have to be reviewed and fine-tuned, but only if the Agile methodology under consideration includes specific values or guidelines. The values of these requirements can then be presented as a subject for a debate between project team members and even customer representatives. It has to be noted that usually most problems of Agile adaptation arise due to inflexibility or stubbornness of project management, personnel, or customer. For example, one of the more common problems is poor customer involvement that is usually due to the passiveness of the customer or his/her unwillingness to invest any effort in the development process. However, majority of Agile methodologies authors argue that such involvement is indeed crucial for the project success. Although some practitioners may oppose this view, we think that if the project team does not meet an environmental Agile requirement, they should try to remove the obstacles or at least improve some aspects of the issue instead of allowing the external conditions take over. As the author of Agile Modeling, Scott Ambler has told one of the authors of this chapter in a private discussion that in aforementioned cases “you need to invest some time making your clients understand the importance of their role in the project and why they need to be actively involved.”

4.2 Classifying Agile Methodology Elements

Various Agile methodologies use different definitions and terms for their elements. They may include such notions as values, properties, strategies, techniques, practices, steps, tasks, process, milestones, cycles, phases, builds, work products, artifacts, documents, architecture, software elements, roles, producers, actors, and team. From the practical point of view, during the adaptation, there is not much

benefit to be gained from decomposing Agile methodologies into elements of so many types. We propose to distinguish only the main elements such as work units (tasks, practices, techniques), producers (roles, teams), and work products (documents, models, software items) that are at the heart of any software development methodology. We also propose to distinguish only those elements that are specific or unique to the Agile ISD approach (as compared with plan-driven approaches). There is also a danger of mistakenly identifying environmental agility requirements as methodology elements. However, if we managed to identify a full set of requirements, it is safe to say that remaining “bits” of the methodology that can be applied individually are its elements that should be extracted during this step.

4.3 Identifying the Levels of Adaptability

Each element of the Agile methodology is usually accompanied by some sort of description of how it should be applied or implemented. However, in real life due to various limitations, specific Agile elements can often be applied only to a certain limited extent. For example, there may be the declarative need for such specific roles within the project as coach (XP) or business expert (Crystal). In reality, however, it may be impractical or financially problematic to hire an external training expert. Maybe the creation of an additional full-time position would put unacceptable burden on the project budget. On the other hand, the part-time consulting on some specific issues may also yield positive results while demanding much less financial resources.

In other words, there is always the need to balance between striving toward best possible result and financial capabilities. With this problem in mind we propose to define the application of each methodology element with regard to three levels of adaptation (see Table 2 for the descriptions of the levels).

4.4 Relating Agile Elements with Concern Areas

Majority of methods and methodologies are driven by certain fears. For example, XP is built around the concern that too much time will be spent on documentation and too little on actual programming. Heavyweight and plan-driven methodologies

Table 2 Adaptability levels of Agile elements

Levels	Basic	Intermediate	Advanced
Description	Define the minimal acceptable amount of element adaptation	Define the average level of element adaptation that requires acceptable efforts	Define the full element adaptation for use in any conditions

Table 3 Concerns/problem areas

Building team	Customer satisfaction/feedback	Project progress visibility
Team knowledge sharing	New technology challenges	Requirements elicitation
Person motivation	Architecture and design	Requirements prioritization
Customer involvement	Effort estimation	Code, design quality
Testing	Work conventions	Staff training

try to tackle the fair of uncontrolled projects, leaving team members, and loss of accumulated knowledge. From the practical point of view, there are always things that people on real projects are most concerned about when deciding what Agile elements to adapt. We have distinguished the set of concerns/interest/problem areas that are presented in Table 3.

Relating concern and problem areas with the Agile elements provides us with a useful new perspective on both our fairs and the methodology elements that can tackle them. A detailed example is given in Table 7.

4.5 Adaptation of the Agile Methodology (Guidelines)

We propose to follow the guide consisting of four steps when adapting Agile methodology for a specific situation:

- [step 1] *Reassessing environmental requirements.* Reassess project environment with respect to the Agile environmental requirements. It is important in order to better understand the risks associated with Agile adaptation. Use team discussions to determine if your environment is ready for Agile and if not, what could be adjusted or changed. Only this way one can expect best results from adopting Agile methodologies.
- [step 2] *Selecting Agile methodology elements with respect to concerns.* Pick up relevant Agile elements that are most likely to solve or lighten existing problems or risks. Prioritize selected elements according to their importance. Use prioritization techniques (e.g., AHP technique) if needed.
- [step 3] *Assessing selected Agile elements with respect to the adaptability levels.* Decide how much of agility you can afford considering the effort you can put in. Use values of basic, intermediate, and advanced levels.
- [step 4] *Implementing elements into a process.* Selected and evaluated Agile elements should be implemented into existing project process. As stated earlier, we believe that companies usually have their own know-how and do not want to rebuild processes from scratch, looking instead for ways on how to improve existing processes and their environment. Specifics of this particular step are not the subject of this chapter and need further research.

5 Case Study – Adapting Crystal Clear

Crystal Clear is a member of the Crystal family of methodologies as described by Alistair Cockburn and is considered an example of an Agile or lightweight methodology. Due to the lack of interest from researchers, Crystal methodologies reside below other popular Agile methodologies such as XP or Scrum and there are not so many sources about their adaptation in literature. Major sources about Crystal Clear and the aspects of its adaptation are presented by the author of the methodology [13–16] or in the Agile Manifesto [3] which was cowritten by the author of this methodology. Most of the other sources are various reviews [1, 2, 32] or cover just some aspects of the classification [31] and are instantiated from the Cockburn’s sources. We used all these sources for the analysis and creation of the adaptation of Crystal Clear.

Crystal methodologies are based on observations of many successful teams and those observations give Crystal a sounder base than most competing methodologies, which typically trace their roots back to a smaller number of projects [33]. We chose Crystal Clear for the case study because of two reasons. First, Crystal Clear is a part of the work that summarizes 10 years of research of successful projects that was done by the author of the methodology. Also, as we already mentioned, there is a lack of formal research on the adaptation of Crystal Clear. We present the results of the adaptation that was carried following the process framework presented in the previous section of this chapter.

- [step 1] *Eliciting environmental requirements.* Most of the values of the environmental Agile requirements for Crystal Clear are the same as those presented in Table 1. However, some requirements are described in more detail in Crystal Clear (see Table 4).
- [step 2] *Classifying Crystal Clear elements.* Identified and classified Crystal Clear elements are presented in Table 5.

Table 4 The environmental requirements of Crystal Clear (descriptions adapted from [16])

Requirement	Description
Project size	Up to eight developers (with exception of the extension to 12 people)
Criticality	Loss of comfort or loss of discretionary moneys. Can be shaped, with additional testing, verification, and validation rules, up to “essential” moneys. Not intended for life-critical systems as it is missing verification of correctness
Software type	Mainframe, client-server, Web-based, using any type database, central or distributed. Can be shaped for hard real-time systems with additional rules for planning and verification of system timing issues. Not intended for fail-safe systems, as it is missing hard architectural reviews for fault tolerance and fail-over
Personnel	One colocated team. Especially strong on communications. Short, rich, informal communications paths, including with the sponsoring and user communities. Tolerance for variations in people’s working styles

Table 5 Elements of Crystal Clear

Type	Elements
Work units	Methodology shaping (MS), Osmotic communication (OC), Frequent delivery (FD), Focus (F), Automated tests (AT), Configuration management (CM), Frequent integration (FI), Exploratory 360° (E), Early victory (EV), Walking skeleton (WS), Code and Design refactoring (CDR), Information radiator (IR), Reflection workshops (RW), Blitz planning (BP), Delphi estimation (DE), Daily stand-ups (DS), Essential interact. Design (EID), Side-by-Side programming (SSP), Process miniature (PM), Team structure and conventions (TSC)
Work products	Mission Statement with Trade-off Priorities, Actor-Goal List, Use Cases and Requirements File, User Role Model, Architecture Description, Screen Drafts, Common Domain Model, Design Sketches and Notes, Source Code, Migration Code, Tests, Packaged System, Project Map, Release Plan, Project Status, Risk List, Iteration Plan and Status, Viewing Schedule, Bug Reports, User Help Text, Team Structure and Conventions, Reflection Workshop Results, Burn Charts
Producers	Executive sponsor, Expert user, Business expert, Lead designer, Designer – programmer, Coordinator, Tester, Writer

[step 3] *Identifying the levels of adaptability.* Due to chapter-length restrictions we present the adaptability levels only for Crystal Clear work units (Table 6).
[step 4] *Relating Agile elements with concern areas.* As with the previous step, due to chapter-length restrictions, we only present several defined relationships between work units and concern areas (Table 7).
[step 5] *Adaptation of the Crystal Clear methodology.* There are always some exceptions when describing adaptation process. The proposed process guide should be tailored according to the specific issues that may arise with the Agile methodology. In Crystal Clear’s case the process guide for adaptation is the same:

- [step 5.1] Assess Environmental requirements.
- [step 5.2] Select Agile methodology elements with respect to concerns.
- [step 5.3] Assess selected Agile elements with respect to the adaptability levels.
- [step 5.4] Implement elements into a process.

Table 6 Adaptation levels of Agile work units (first cell – technique (T))

T	Basic	Intermediate	Advanced
MS	Do project interviews and methodology reshapes at least after the projects	Review and reshape methodology after each delivery if needed	Do project interviews during iterations and reshape methodology if needed
OC	Organize osmotic meetings as often as possible	One colocated team, face-to-face communication	One colocated team, face-to-face communication over documentation

Table 6 (continued)

T	Basic	Intermediate	Advanced
F	Provide every team member enough time only for value-critical tasks	In addition, provide enough time to the key team members	Provide every team member enough time and space for focusing on current tasks
AT	Run tests for complex program units and during integration	In addition, run tests for code units that were untested during several increments	Run tests for every program unit daily and during integration
CM	Use tools that allow check in locally	Use tools that allow check in over the Internet	Use tools that allow check in over the Internet
FI	Every month	Every 2 weeks	Every week
EV	Do easy things first if ambiguity of the hard ones is low	Do easy things with respect to importance first on every iteration	Balance between doing easy things first and last with respect to current motivation of developers
WS	Build whole prototype of working system	In addition, implement several working functions	Build whole working system with a minimum set of working functions
CDR	Only where reuse, complexity or integrality is high	In addition, do refactoring that was not done during several increments	Everywhere as needed during each iteration
IR	Use verbal status reports on meetings, send status mail	Use status board or monitor visible to all team members	Use a specific or adopted network/Internet project status reports tools
RW	Hold reflection workshops at least after the project	After each delivery revise working conventions	Do reflection workshops every time when needed
BP	Use index cards for rapid task assessment, create several plans	Involve customer representative during planning	Use full technique and create project plans not longer than 3 months
DE	Estimate using use cases	Estimate using use cases and UI screens	Combine using business classes, use cases or UI screens for estimation
DS	Only most problematic developers report their problems several times a week	All team members report about their problems and status every day	In addition, team members not only report but discuss their problems and help each other
SSP	Use one room without partitioning walls	Use Side-by-Side programming for development	Combine with pair-programming, put expert developers side-by-side with junior programmers
PM	Capture only the core methodology elements into a material	Create full project methodology overview material for training	In addition, renew if methodology reshape was done
TSC	Create and use team structure and working conventions for project	In addition, change work conventions during reflective workshops	In addition, change and reuse anytime if needed

6 Concluding Remarks

The problem of adapting Agile methodologies or their elements to a specific project environment is still quite sharp as the availability of formal approaches is very limited. In this chapter, we proposed the process of preparing for the Agile methodology adaptation as well as described the guidelines of the adaptation based on the preparation results.

The proposed process framework utilizes several novel ideas. First is the definition of 13 environmental Agile requirements that have to be met in order to successfully implement Agile methodology or its elements. We also showed that situational method engineering techniques may not be very effective in case of Agile adaptation. It is enough to decompose such methodologies into elements of only three types: producers, work units, and work products. In addition to that we proposed defining not one but three possible adaptation levels of each Agile element (basic, intermediate, and advanced). As part of the preparation for adaptation process we have suggested relating such elements of a specific Agile methodology to a defined set of 16 areas of concern. This matrix, along with other proposed analysis results, provides decision makers with an ultimate set of aids to evaluate the chosen Agile methodology and proceed with its (partial) implementation into the existing ISD processes.

The proposed methodology proved to be relatively easily applicable and very useful during the simulation of the preparation for the adaptation of the Crystal Clear methodology. The case study also revealed some possible shortcomings of the approach, namely, the need for a lengthy methodology elements evaluation tables. However, these evaluations (like the ones produced during the case study) can be successfully reused and do not need to be built from scratch every time.

Acknowledgment This work is supported by the Lithuanian State Science and Studies Foundation according to the High Technology Development Program Project “VeTIS” (Reg. No. B-07042).

References

1. Abrahamsson, P., Salo, O., Ronkainen, J., and Warsta, J. (2002) Agile Software Development Methods: Review and Analysis, VTT Publications.
2. Abrahamsson, P., Warsta, J., Siponen, M. K., and Ronkainen, J. (2003) New directions on Agile method A comparative analysis. In: Proceedings of the 25th International Conference on Software Engineering, IEEE Computer Society, pp. 244–254.
3. Agile Alliance (2001) Principles behind the Agile Manifesto. Retrieved 14 May, 2009, from: <http://agilemanifesto.org/principles.html>.
4. Ambler S. W. (2002) Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process. John Wiley & Sons.
5. Ambler S. W. (2004) The Object Primer, 3rd Edition. Cambridge University Press.
6. Ambler S. W. (2007) Agile Adoption Rate Survey: March 2007. Retrieved 15 May, 2009, from: <http://www.ambysoft.com/downloads/surveys/AgileAdoption2007.ppt>.
7. Attarzadeh, I. and Hock, O. S. (2008) New direction in project management success: Base on smart methodology selection. In: Proceedings of Information Technology Symposium 2008, Vol. 1, pp. 1–9.

8. Beck K. (2004) *Extreme Programming Explained: Embrace Change*, 2nd Edition. Addison Wesley Professional.
9. Beck, K. (1999) *Extreme Programming Explained: Embrace Change*. Addison-Wesley.
10. Boehm, B. (2002) Get Ready for the Agile Methods, with Care. *Computer*, IEEE Computer Society Press, Vol. 35(1), pp. 64–69.
11. Boehm, B. and Turner, R. (2003) Using Risk to Balance Agile and Plan-Driven Methods. *Computer*, IEEE Computer Society Press, Vol. 36(6), pp. 57–66.
12. Boehm, B. and Turner, R. (2004) *Balancing Agility and Discipline*. Addison-Wesley.
13. Cockburn, A. (1998) *Surviving Object-Oriented Projects: A Manager's Guide*. Addison Wesley.
14. Cockburn, A. (2000) Selecting a Project's Methodology. *IEEE Software*, IEEE Computer Society Press, Vol. 7(4), pp. 64–71.
15. Cockburn, A. (2002) *Agile Software Development*. Addison-Wesley.
16. Cockburn, A. (2004) *Crystal Clear: A Human-Powered Methodology for Small Teams*. Addison Wesley Professional.
17. Danikauskas, T., Butleris, R., Drasutis, S. (2005) Graphical user interface development on the basis of data flows specification. In: *Proceedings of the 20th Computer and Information Sciences Symposium*. Berlin: Springer, *Lecture Notes in Computer Science*, Vol. 3733, pp. 904–914.
18. Drobka, J., Nofzt, D., and Raghu, R. (2004) Piloting XP on Four Mission-Critical Projects, *IEEE Computer*, IEEE Computer Society Press, Vol. 21(6), pp. 70–75.
19. Firesmith, D. G. and Henderson-Sellers, B. (2002) *The OPEN Process Framework. An Introduction*. Addison-Wesley.
20. Georgiadou, E., Siakas, K. V., and Berki, E. (2007) Agile quality or depth of reasoning: Applicability versus suitability respecting stakeholders' needs. In: *Agile Software Development Quality Assurance*. Information Science Reference, pp. 23–55.
21. Greer, D. and Ruhe, G. (2004) Software release planning: An evolutionary and iterative approach. *Information and Software Technology*, IEEE Computer Society Press, Vol. 46(4), pp. 243–253.
22. Henderson-Sellers, B., Gonzalez-Perez, C., and Ralyte, J. (2008) Comparison of Method Chunks and Method Fragments for Situational Method Engineering. *Software Engineering ASWEC 2008*, IEEE Computer Society, Vol. 18(6), pp. 479–488.
23. ISO/IEC. (2007) ISO/IEC 24744, *Software Engineering. Metamodel for Development Methodologies*. International Standards Organization/International Electrotechnical Commission.
24. Jeffries, R. E., Anderson, A., and Hendrickson, C. (2000) *Extreme Programming Installed*, Addison-Wesley.
25. Kumar K. and Welke R. J. (1992). Method engineering: A proposal for situation specific methodology construction. In: *Systems Analysis and Design: A Research Agenda*. Cotterman, W. W., Senn, J. A. (Eds). Wiley, pp. 257–268.
26. Lan, C., Mohan, K., Xu, P., and Ramesh, B. (2004) How Extreme Does Extreme Programming Have to Be? Adapting XP Practices to Large-Scale Projects. In: *Proceedings of the 37th Hawaii International Conference on System Sciences*, IEEE Press, Vol. 3, pp. 342–250.
27. Layman, L., Williams, L., and Cunningham, L. (2004) Exploring extreme programming in context: An industrial case study. In: *Proceedings of the Agile Development Conference*, IEEE Computer Society, pp. 32–41.
28. Mikulenas, G. and Butleris, R. (2009) An approach for modeling technique selection criteria. In: *Proceedings of the 15th International Conference on Information and Software Technologies, IT 2009*, Kaunas University of Technology, pp. 207–216.
29. Mirbel, I. (2006) Method chunk federation. Workshop on Exploring Modeling Methods for Systems Analysis and Design – EMMSAD'06, held in conjunction with the 18th Conference on Advanced Information Systems Engineering – CAISE 2006, Namur University Press, pp. 407–418.

30. OMG. (2002) Software Process Engineering Metamodel Specification, formal/2002-11-14. Object Management Group.
31. Quynh, N. T., Henderson-Sellers, B., and Hawryszkiewicz, I. (2008) Agile method fragments and construction validation. In: Handbook of Research on Modern Systems Analysis and Design Technologies and Applications. Rahman, S. M. (Ed). Idea Group, Inc., pp. 243–271.
32. Ramsin, R. and Paige, R. F. (2008) Process-centered review of object oriented software development methodologies. ACM Computer, 40(1):1–89.
33. Rusk J. (2009) Crystal Clear Methodology. Retrieved 21 May, 2009, from: http://www.agilekiwi.com/crystal_clear.htm.
34. Schwaber K. (2004) Agile Project Management with Scrum. Microsoft Press.
35. Silingas, D. and Butleris, R. (2008) UML-intensive framework for modeling software requirements. In: Proceedings of the 14th International Conference on Information and Software Technologies IT 2008, Kaunas University of Technology, pp. 334–342.

The Methodology Evaluation System Can Support Software Process Innovation

Alena Buchalcevova

Abstract This chapter focuses on an evaluation of software development methodologies and a selection of the appropriate methodology for a concrete project. Seeing that the present status of using development methodologies worldwide and especially in the Czech Republic is not satisfactory, the Methodology Evaluation System METES was created. The METES system is based on common criteria for the methodology selection, but it also takes into account specific conditions of software development in the Czech Republic. The METES system was used for the evaluation of six selected present methodologies.

Keywords Methodology · Information system · Rigorous methodology · Agile methodology · Methodology evaluation · Methodology selection

1 The Status of Software Development in Present Days

In the present turbulent world, rapid changes in the economic environment are under way and more distinctive changes occur in the area of information systems and information and communication technologies (IS/ICT). Today's economic climate forces companies to measure their results according to revenue, costs, and quality. Companies have to focus on those projects with an immediate value to the business. IT projects under this pressure have to be made right the first time, on time, and match to customer requirements.

According to the Standish Group's research conducted in 2006 [8], only 35% of all application development projects satisfied criteria of successfulness (project finished on time, according to the budget, and with all specified functions). Based on the research results the Standish Group has defined 10 key success factors for IS/ICT projects. The main reasons why projects fail are low level of user involvement, low

A. Buchalcevova (✉)

Department of Information Technologies, University of Economics, Prague, Czech Republic
e-mail: buchalc@vse.cz

management support, clear business objectives, project scope optimization, agile processes, project management skills, and formal methodology.

According to the research done in 2006 in the Czech Republic [5], 5% of software development companies do not use any methodology, 14% use only company standards, and additional 5% use agile methodologies. Using the appropriate methodology for a concrete project represents another problem. Different methodologies are needed depending on the project size (number of people being coordinated), the criticality of the systems being created, and the priorities of the project. Traditional or rigorous methodologies do not suit the projects with changing requirements, where on the other hand agile methodologies bring better results.

With an aim to support companies in the methodology selection and customization process we defined the Methodology Evaluation System METES described in Section 3.

2 Existing Systems for Methodology Evaluation and Selection

Supporting the process of selection the appropriate methodology is not the latest phenomenon. There are several such systems nowadays. In this section some of them are analyzed. First mentioned is the Methodology Framework for IS/ICT – MeFIS presented in [3, 4]. MeFIS defines methodology patterns for various project types, problem domains, and solution types (new IS development, upgrade IS, package implementation), and specifies principles and processes for the selection of an appropriate pattern and its customization to the conditions of a concrete project. From the methodology selection point of view especially the specification of the criteria for categorization of methodologies is to evaluate. However, the process of methodology selection was not developed in detail in the MeFIS framework.

Hecksel defined the System and Method for Software Methodology Evaluation and Selection which was accepted as a patent [7]. Each project for which a methodology is selected has its Project Context consisting of multiple components including People, Process, and Technology. Each component has multiple attributes whose values are compared to the methodology models using simulation, statistics, and statistical forecasting. Hecksel's system has been well worked out and supported by a software tool. In my opinion, this system containing 62 attributes is too complex but on the other hand some attributes like methodology availability, support, and localization are not included.

Boehm and Turner [2] present a set of conditions which are important in a decision in using either agile or plan-driven methods, i.e., project criticality, project size, quality of people, dynamism, and company culture. These criteria are definitely core and should be included in any methodology selection system. Boehm and Turner's system was then modified by Taylor who added the client involvement criterion [12]. Furthermore, on the basis of the case studies results he replaced the company culture criterion by the team distribution criterion [6].

3 The Methodology Evaluation System METES

The Methodology Evaluation System METES comprises both an assessment system and a basis of assessed methodologies we can choose from. Figure 1 shows the conceptual model of METES in the form of UML 2.0 class diagram.

The structure of the assessment system is captured in Fig. 2. Each methodology is first shortly described and then values for evaluation criteria are assessed. The evaluation criteria are clustered into four groups – Process, Support, Product, and People. Criteria in the Process group represent process features of the methodology, e.g., scope of software life cycle processes, life cycle process model, roles, metrics, and type of development. Criteria included in the Support group assess availability of the methodology, support of the implementation and customization of the methodology, availability of skilled people, etc. In the Product group there are criteria evaluating the built solution. Criteria in the People group describe features of the development team.

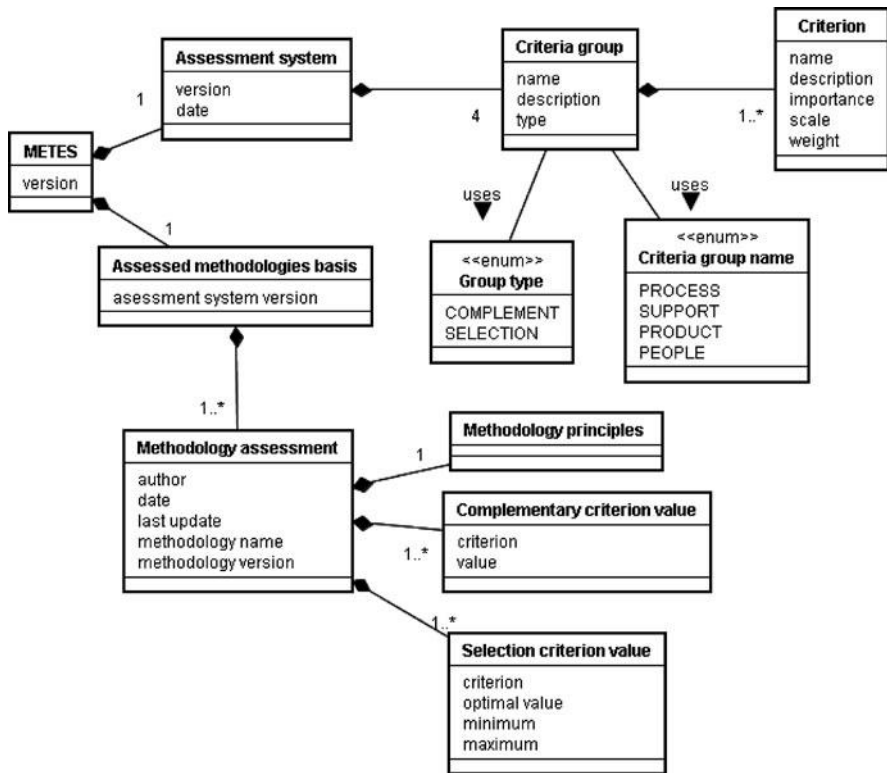


Fig. 1 The conceptual model of the METES system (resource: author)

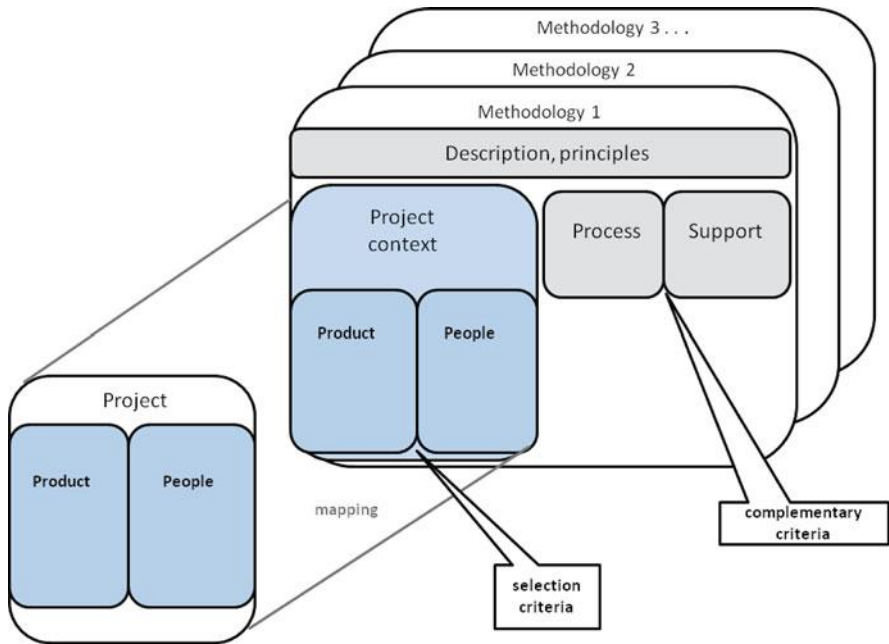


Fig. 2 Structure of the METES system (resource: author)

The criteria groups Product and People represent project context which means that values of this criteria are assessed for the project and mapped to the criteria values of individual methodologies. In this case these criteria represent selection criteria. Among the selection criteria, five key criteria (see Section 3.1) were identified. These criteria are, in opinion of the author, the most critical for the selection of the methodology. The key criteria are the same as proposed by Boehm, Turner, and Taylor but in addition Project duration criterion is inserted. Criteria in the Process and Support groups compose complementary criteria that are used as an additional tool in the selection process.

For each criterion, a scale from 0 to 5 is defined, where in general 0–1 means low level of compliance, 2–3 medium level of compliance, and 4–5 high level of compliance. For each criterion, a detail meaning of values on a scale is determined. Detailed description of all criteria and their role in METES exceeds the scope of this chapter; therefore, only a short explanation follows.

3.1 Key Selection Criteria

Product criticality: The basic criterion which differentiates individual software projects is criticality of the designed system. This criterion assesses if the system is just for entertainment (value 1), mission supported (value 2), mission critical at

the national level (value 3), mission critical at the global level (value 4), or life critical (value 5).

Project duration: Duration of the project is measured in months (0 = less than a month, 5 = more than 24 months). In the present turbulent world, the key factor of project success is time to market. In general, for short-term projects light methodology is suitable.

User accessibility: Customer involvement in the project is one of the key criteria for methodology selection. Implicit assumption for using agile methodologies is that the user (customer) will be accessible daily or better a direct part of the team. Value 0 is assigned when the user is a part of the team and is responsible for requirements. Value 5 is assigned where the user is not accessible during the project.

Team size: Size of the team measured usually by number of its members belongs to the key criteria for methodology selection. More people in a team require more communication and more formal methodology. Agile methodologies are intended for small colocated teams. Value 0 is assigned for less than 4 team members. Value 5 is assigned for more than 100 team members.

Distribution: Distributed teams have recently become an important part of software development. Agile methodologies in the basic concept do not suit distributed teams. This criterion specifies if the team is located in one room (value 0), one city (value 2), one country (value 4), or in various countries (value 5).

3.2 Other Selection Criteria

Requirements stability: Value of this criterion expresses extent of requirements changes during the project. Value 0 is assigned when requirements cannot be defined ahead. Value 5 is assigned for project with none or few changes of requirements.

Reuse: The Reuse criterion determines extent of usage or building reusable artifacts in the project. Agile methodologies as it is known are not suitable for building reusable software components. Value 0 is assigned for no reuse (better said as nontargeted reuse). Value 5 is assigned for project focused on building reusable components at the enterprise level.

Solution size: This criterion measures the size of the solution (system) by number of use cases. Value 0 is assigned for less than 10 use cases. Value 5 is assigned for project with more than 300 use cases.

Lack of project manager experience: This criterion shows how experienced a project manager does the methodology require. Value 0 is assigned for requirement of more than 5 years of project manager experience. Value 5 is assigned for less than 1 year of project manager experience.

Lack of team member qualification: This criterion reflects level of qualification of team members that the methodology supposes. Agile methodologies, for example, work properly when most of the team members are highly qualified. Value 0 is assigned for requirement of more than 70% team members with good qualification. Value 5 is assigned for more than 80% low-qualified team members.

Lack of team member motivation: This criterion evaluates how motivated team members are or should be. Value 0 is assigned for very motivated staff with high moral values. Value 5 is assigned for low or no motivation.

3.3 Complementary Criteria

Scope: The Scope criterion assesses the number of software life cycle processes covered by the methodology. The evaluation is based on mapping to the Process Reference Model ISO/IEC 12207.

Life cycle model: This criterion assesses the life cycle model (see definition in [11]) used by the methodology. The highest value is assigned for iterative model with iteration length less than 1 month.

Role: The Role criterion assesses the number of roles the methodology deals with. Value 5 means that both software engineering and management roles at the project and enterprise level are represented in the methodology.

Process description particularity: This criterion assesses the level of detail in a process description. The highest value represents complete process description, e.g., inputs, outputs, roles, and tasks performed by roles.

Documentation: The Documentation criterion assesses the amount of documentation that the methodology requires by mapping to documents stated in ISO/IEC 15289.

Metrics: This criterion assesses how the methodology works with metrics and how important metrics are.

Quality management: This criterion assesses the level of testing and quality management incorporated in the methodology.

Integrity of resources: This criterion looks on methodology resources. Some methodologies are accessible from books or various papers and from websites and some are delivered as applications with content management tools (value 5 is assigned).

Availability: This criterion evaluates whether the methodology is available in the form of books, as a commercial product, free, open-source, or open-source with content management tools (value 5 is assigned).

SW tools support: This criterion has two dimensions. First, one looks on software tools supporting activities described in the methodology. Second, one assesses support for methodology content, its publication, and customization.

Methodology implementation support: This criterion assesses the level of support for methodology implementation, e.g., consultation, training, and methodology configuration.

Methodology customization: This criterion assesses how the methodology deals with customization and if methodology customization is possible, is recommended, and is supported by software tools (value 5 is assigned).

University courses: This criterion expresses whether the methodology is taught at universities either theoretically or practically. The highest value is assigned for

teaching practical use of the methodology (in labs) at universities in the Czech Republic.

Training and certification: This criterion assesses availability of training and certification either worldwide or in the Czech Republic (value 5 is assigned).

Localization: This criterion evaluates availability of the methodology in the Czech language.

4 The Evaluation of Selected Present Methodologies

The assessment system of METES was used for the evaluation of some present methodologies. Selected methodologies belong to the most used ones according to the results of several researches focused on usage of software development methodologies [1, 5, 9], i.e., Rational Unified Process (RUP), OpenUP, Feature-Driven Development (FDD), Scrum, Extreme Programming (XP), and MSF for CMMI development. Selection criteria values of all selected methodologies assessed by expert estimation are shown in Table 1.

Results of the assessment can be presented in a graphical form as shown in Figs. 3, 4, and 5. These graphs enable visual comparison of methodologies. From Fig. 3, it is obvious that XP methodology is much more lightweight than RUP methodology and is suitable for smaller colocated teams and less critical products, and supports changes during the project, but on the other hand requires highly qualified and well-motivated people. Figure 4 clearly shows that XP is more lightweight than RUP in the process area too.

Comparing availability and support of RUP and XP (see Fig. 5), we can see high contrast in the integrity of resources criterion. RUP is delivered as one integrated resource (application with the software tool for content management and

Table 1 Values of the selection criteria of selected methodologies

Criterion	RUP	Open UP	FDD	Scrum	XP	MSF CMMI
Product criticality	5	2	3	3	3	5
Project duration	4	2	2	3	2	4
Requirements stability	2	1	1	0	0	3
Reuse	3	2	2	1	1	3
Solution size	5	2	5	5	3	5
Lack of project manager experience	4	4	3	2	2	4
Lack of team member qualification	5	5	3	1	1	5
Lack of team member motivation	4	4	2	1	1	4
User accessibility	4	3	1	0	0	4
Team size	5	2	3	3	1	5
Distribution	5	1	1	3	1	5

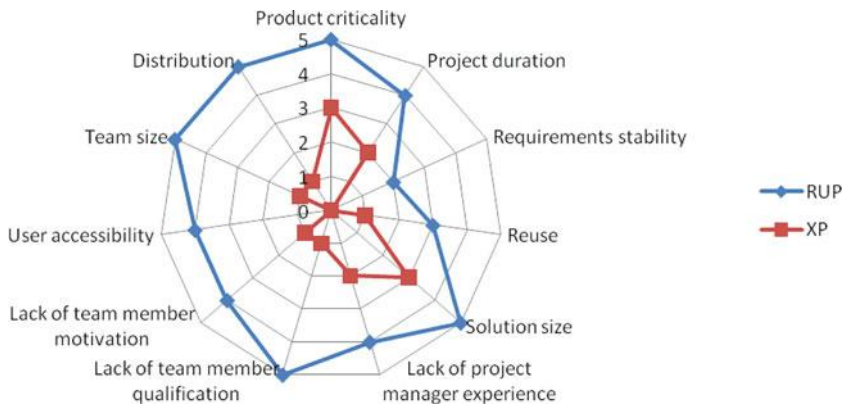


Fig. 3 Comparing values of the selection criteria of RUP and XP (resource: author)

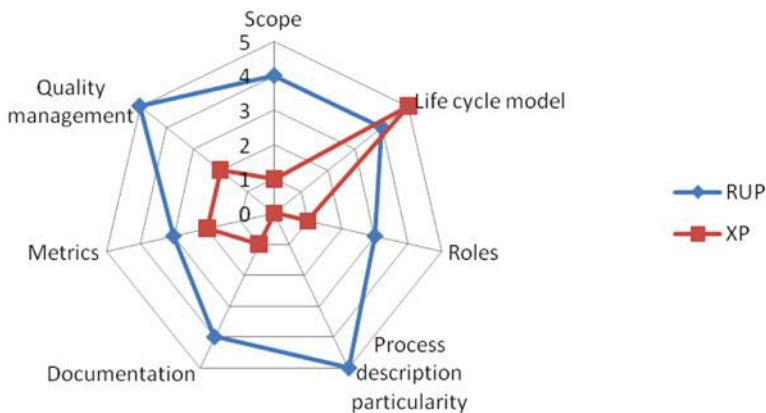


Fig. 4 Comparing values of the Process group criteria of RUP and XP (resource: author)

publication – Rational Method Composer), while XP is available in fragmented resources – books, papers, and web presentations. However, availability of training, certification, and university courses for both methodologies is on the highest level.

5 Process of the Methodology Selection

As the methodology selection method is one of the multicriteria analysis methods, criteria weights are defined based on the Saaty’s method of quantitative pair comparison [10]. Weights are calculated according to expert evaluation of relative preferences among each two criteria. The METES system includes default criteria weights (see Fig. 6), but the company can define its own weights according to specific project characteristics.

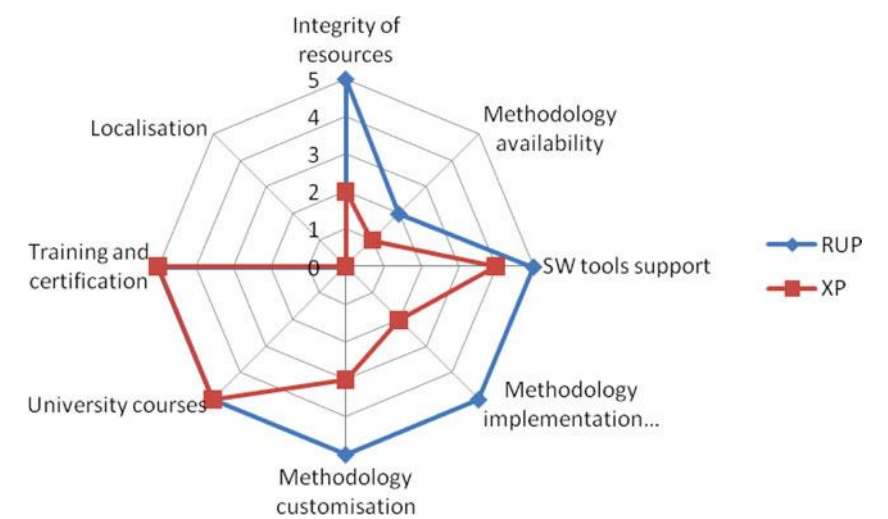


Fig. 5 Comparing values of the Support group criteria of RUP and XP (resource: author)

Selection criteria	weight	Complementary criteria	weight
Product criticality	0,219	Scope	0,051
Project duration	0,133	Life cycle model	0,089
Requirements stability	0,041	Role	0,026
Reuse	0,033	Process description particularity	0,059
Solution size	0,039	Documentation	0,027
Lack of project manager experience	0,015	Metrics	0,030
Lack of team member qualification	0,020	Quality management	0,038
Lack of team member motivation	0,020	Integrity of resources	0,195
User accessibility	0,200	Availability	0,195
Team size	0,169	SW tools support	0,106
Distribution	0,113	Methodology implementation support	0,038
	1,000	Methodology customisation	0,038
		University courses	0,023
		Training and certification	0,025
		Localisation	0,059
			1,000

Fig. 6 Criteria weights (resource: author)

The process of the methodology selection is divided into two steps. The aim of the first step is to select applicable methodologies for the project. First, we assess selection criteria values (i.e., criteria in Product and People group) for the project. These project values are compared to the selection criteria values of various methodologies. We select those methodologies for which each project key selection criterion value (product criticality, project duration, user accessibility, team size, distribution) is in between minimal and maximal values for the methodology criterion.

The aim of the second step is to select one or more recommended methodologies from the list of applicable methodologies. To do so we have

- (1) to find the methodology, which has the lowest value of the weighted sum of distances from optimal selection criteria values calculated according to this formula:

$$\sum_{i=1}^{11} (|pv_i - mopt_i| \times vv_i)$$

where pv_i is project selection criteria values; vv_i is selection criteria weights; and $mopt_i$ is optimal values of selection criteria for the methodology.

- (2) to evaluate the complementary criteria, i.e., criteria in the Process and Support group. We can analyze each value of the complementary criterion separately or use the highest value of weighted sum calculated according to this formula:

$$\sum_{i=1}^{15} (md_i \times vd_i)$$

where md_i is values of the complementary criteria for methodology and vd_i is complementary criteria weights.

This selection process was verified on two real running projects; just a brief summary of one of them is stated. The project for which we selected the appropriate methodology was an information system of the Bar Association created by four people in a team within 1 year. As the user was available only in the beginning and at the end of the project, the value of the User availability criterion was outside the minimal and maximal intervals for agile methodologies. Due to the small size of the team, RUP and MSF CMMI were not applicable too. Finally, the methodology to recommend was stated OpenUP.

6 Conclusions

This chapter focuses on the evaluation and selection of software development methodologies. It introduces the original system for the methodology evaluation

and selection named METES. In comparison to other systems for methodology evaluation and selection, the METES system enriches the methodology selection process by introducing new criteria such as integrity of resources, availability of the methodology, support of the methodology implementation and customization, occurrence of university courses, training and certification, and localization of the methodology. The METES system was verified through the assessment of six selected presently often used methodologies, both rigorous and agile.

Acknowledgment The work reported in this chapter was supported by the project GA CR 201/08/0663 information systems innovation supporting company competition.

References

1. Ambler, S. W. Agile software development methods and techniques are gaining traction. In: *Dr. Dobb's Portal*, August 2006 [online]. Think Services, c2007 [cit. 2007-10-06]. WWW: <<http://www.ddj.com/dept/architect/191800169>>.
2. Boehm, B.; and Turner, R. *Balancing Agility and Discipline: A Guide for the Perplexed*. Canada: Addison-Wesley Professional, 2004. ISBN 0-321-18612-5.
3. Buchalceva, A. Methodology Patterns. Orlando 21.07.2004–25.07.2004. In: Chu, Hsing-Wei, Aguilar, Jose, Ferrer, Jose, Syan, Yu-Ru, Cheng, Chin-Bin (eds.). *CITSA 2004*. Orlando: IIS, 2004, s. 200–203. ISBN 980-6560-19-1.
4. Buchalceva, A. Application of Object-Oriented Principles to the Methodology Area. Karlstad 14.08.2005 –17.08.2005. In: Nilsson, A. G., Gustas, R., Wojtkowski, W., Wojtkowski, W. G., Wrycza, S., and Zupančič, J. (eds.). *ISD'2005*. Karlstad: Karlstad University, 2005, pp. 49–57. ISBN 91-85335-72-X.
5. Buchalceva, A. Research of the Use of Agile Methodologies in the Czech Republic. In: Barry, C., Lang, M., Wojtkowski, W., Wojtkowski, G., Wrycza, S., and Zupancic, J. (eds.). (2008.) *The Inter-Networked World: ISD Theory, Practice, and Education*. New York: Springer, ISBN 978-0387304038.
6. Caffery, F.; Taylor, P.; and Coleman, G. Adept: A Unified Assessment Method for Small Software Companies, In: *IEEE Software*. Vol. 24, Issue 1, Jan.–Feb. 2007, pp. 24–31 [online] [cit. 2007-10-07]. <<http://ieeexplore.ieee.org/iel5/52/4052534/04052548.pdf>>.
7. Hecksel, D. L. *Methodology Evaluation and Selection*. [online] Sun Software Services, Sun Microsystems [cit. 2007-10-07]. WWW: <<http://www.davidhecksel.com/projectcontext/whitepaper.html>>.
8. Johnson, J. *My Life Is Failure*. The Standish Group International, Inc. USA, 2006. ISBN 1-4243-0841-0.
9. Larsen, D. *Agile Alliance Survey: Are We There Yet?* [online] C4Media, Inc., USA, c 2006–2007 [cit. 2007-10-06]. WWW: <<http://www.infoq.com/articles/agile-alliance-survey-2006>>.
10. Saaty, T. L. Relative Measurement and Its Generalization in Decision Making: Why Pairwise Comparisons are Central in Mathematics for the Measurement of Intangible Factors – The Analytic Hierarchy/Network Process. *RACSAM (Review of the Royal Spanish Academy of Sciences, Series A, Mathematics)* 102(2): 251–318. <http://www.rac.es/ficheros/doc/00576.PDF>.
11. *Software and Systems Engineering Vocabulary* [online] IEEE Computer society, c2008 [cit. 2008-01-30]. WWW: <http://pascal.computer.org/sev_display/index.action>.
12. Taylor, P. S. Applying an Agility/Discipline Assessment for a Small Software Organisation, *Product Focussed Software Process Improvement, 7th International Conf. PROFES 2006*, LNCS 4034, Springer Berlin/Heidelberg 2006, pp. 290–304.

Index

A

Agent, 3, 46, 73–74, 190, 194–197, 233–239, 243–253, 335, 345–355, 357–365, 372, 487
 -oriented software engineering, 243–253
 Agile methodology, 486–487, 489, 549, 551–553, 555
 Agile methods, 43, 450, 465–466, 471, 473, 485–486, 500, 548–549
 Agile practices, 451, 453, 485–495, 548–549
 Agile software development, 45, 165, 450, 486
 Agility indicator, 449–457
 AHP, 156, 476, 489, 495, 553
 Architecture of system, 373
 Assessment model, 333–342
 Autonomous team, 451

B

B2B e-business, 367–375
 B2C e-business, 358, 361–364
 Bayesian, 335, 377–380
 Business reshaping, 18, 19–20, 23

C

Call management centre (CMC), 345–347, 349, 351
 Change
 management, 55–64, 119, 152, 455–456, 540
 patterns, 56, 59–62, 401
 CMML, 135–136, 139, 142, 145–146, 567, 570
 Colony's utility, 425, 428
 Component-based development, 282
 Conjoint analysis, 425, 433
 Constraint, 74, 79, 124, 151, 154, 157, 227, 259, 261, 270–271, 273–275, 282, 284, 310–311, 352, 465, 468–469, 473, 523–558

Context sharing, 451, 453, 456

Control, 16–17, 23, 27–31, 34, 43, 56–58, 60, 63–64, 70, 78, 82, 89, 112, 114, 126, 128, 131, 135–145, 152–153, 163–165, 171, 179, 182, 206, 208, 226–227, 231, 235, 239, 257–258, 263, 290, 303, 310, 346, 360, 362, 370–374, 378, 389, 416–417, 420, 452, 454, 464, 468–469, 471, 494, 500–501, 513, 524, 527–532, 536–538, 540

Cost

-effective evaluation approach, 175–184
 -value estimation, 486, 494

Credit, 94, 333–342

Crystal Clear, 554–557

D

Data

migration, 270, 274–278
 quality, 163, 214–219

Development, 17, 20–24, 28–29, 31–33, 41–51, 68, 77–83, 86–88, 92, 103, 106–107, 113–114, 124, 129, 135–146, 151, 153, 155, 157, 166–171, 177, 187–188, 190, 193–194, 202, 211, 213–221, 223–231, 243–246, 248–251, 255–257, 263–264, 269–270, 281–283, 284, 285–288, 297–299, 303, 307–311, 314–315, 329, 346, 354, 358–359, 365, 367–375, 411, 421, 426–428, 430, 432, 438, 440, 442, 445–452, 461, 464, 466–471, 485–488, 492, 501, 503, 505–506, 509, 511–521, 523, 533, 535–537, 540–542, 547, 549–551, 556, 561–563, 567

Dynamic inventory, 377–382

E

Early user-testing, 499–509
 e-Commerce, 223, 244, 246, 333–342, 383
 Educational institution, 187–188, 190, 193–194, 197
 e-Government systems development, 225
 e-Municipality, 166–169
 Enterprise 2.0, 163–165
 Enterprise
 architecture, 42, 57–58, 123–132, 161–163, 320, 328
 informatization, 77–83
 systems, 15–24, 41–52, 319–320
 Environmental management, 124–126, 129–131
 Experimental study, 175, 177, 412, 415–421
 Express, 106, 108, 139, 203, 274, 351, 383–393, 412, 440, 479, 542

F

Factor analysis, 428

G

GAMP, 536–537, 539–542, 544
 Guideline inspection, 178, 180, 182–183

I

Ideal point, 475–481
 Industrial standards, 189–193, 196
 Information systems, 16, 19, 21, 24, 27–38, 41, 43–45, 51, 55, 62, 67–75, 77–83, 85–86, 116–118, 123–131, 145, 162–163, 165–170, 188, 190, 206, 213, 223–225, 269–278, 281, 298, 445–446, 450, 518, 547, 549–550, 561, 570
 Instantiation, 30, 248, 253, 256, 523–533
 Instructional design, 86, 92
 Interorganizational relationships, 535–545
 i* Requirement models, 346, 351–354
 IS management, 130
 ISO, 69, 73, 125–126, 161, 257, 300–302, 406, 535, 537, 550, 566
 IT artifacts, 18, 118, 260, 311–313, 524, 525–526, 530–531, 533, 550–551, 565

J

JADE, 358, 360–361, 364

K

Knowledge
 management, 79, 81, 201–203, 208, 233–239
 requirements, 187–197
 tree, 201–211

L

Levels of change, 15, 18, 23–24
 Life cycles, 18, 20–22, 125–126, 129–131, 287

M

Market forecasting, 383
 MDA, 269, 284, 297, 307, 310, 322–323, 329
 Method
 enactment, 223
 engineering, 486–488, 524, 549
 support, 18, 22, 24
 Methodology, 17, 22, 28, 86, 89–90, 126–131, 154, 165, 175–176, 178–179, 224–225, 228–229, 231, 244–245, 247–248, 252, 256, 288, 297, 303–304, 308–312, 315, 427, 466, 486, 488–489, 548–549, 551–557, 561–571
 evaluation, 561–571
 selection, 488, 562, 565, 568–570
 Model-driven, 255, 269–278, 284, 295–304, 307–316, 323
 development, 255
 Model evolution, 270–276, 278
 Modeling, 28–32, 35, 85–92, 116, 257, 265, 270, 278, 299–300, 307, 411–413, 415–417, 419, 421, 427, 487, 489, 524, 526, 547, 550–551
 Multi-agent, 233–239, 244, 248, 346, 348–351
 Multi agent system (MAS), 235–236, 244, 248, 346, 348–351

N

Negotiation, 100, 105, 223, 244, 246–247, 349, 351, 358, 362–363, 469, 540, 544, 548, 550

O

Object modeling, 85–95, 270
 Outsourcing, 18, 135–142, 145, 345, 402–403

P

Pearson correlation, 388
 Petri nets, 28, 31–32
 Prioritization, 454, 456, 486, 488–491, 553, 557
 Problem-solving methods, 243–253

Process-oriented information systems, 27–38

Project

-based learning, 99–109

management, 43, 49, 111–113, 115, 126, 135, 137, 150, 152–154, 156–157, 165, 170, 190, 192, 227, 402, 405, 455, 464, 470–471, 488, 492, 536, 551, 562

manager, 34, 35, 43, 101, 103, 106, 108, 111–119, 135–136, 140, 142–146, 193, 228–231, 405, 407, 453–454, 463, 465, 468–471, 524, 527, 539–540, 542, 565, 567, 569

portfolio management, 161–171

Q

Quality

assurance, 446, 535–545

in web engineering, 303

Query, 5–12, 216, 271, 336, 358, 362, 364

R

Reasons, 49, 74, 78, 112, 114, 119, 137, 145, 188, 231, 255, 348–350, 405–409, 505, 554, 561

Regression analysis, 385, 389, 391, 430

Relationship manager (RM), 351

Reputation system, 334–336

Requirements

engineering, 259, 413, 420, 488

management, 140, 223–231, 450, 455

prioritization, 223–224, 228

Rigorous methodology, 562

S

Sales

order processes, 45

portals, 46

Scrum, 138, 405, 449–458, 486, 547, 554

Service

management, 15, 17, 19, 23, 202

science, 15–24

Simulation, 17–18, 30, 32, 37–38, 113, 281, 291, 347, 355, 425, 434, 558, 562

Situated action, 224, 227, 464

SOA, 17–18, 42, 284–285, 320–321, 327, 367–375

Social contract, 99–109

Software

architecture, 140, 162, 256, 258, 264–266, 288

development, 17, 28, 43, 45, 51, 75, 87, 135–146, 154–155, 165, 226, 243, 245, 255–256, 264, 295, 297,

308, 371, 399–409, 411, 414, 417, 450–452, 455–457, 462–463, 466–467, 469, 486–487, 499, 505, 523, 533, 541, 547, 549, 551–552, 561–562, 565, 567, 570

metrics, 17–18, 137, 156

process, 27–38, 31–38, 149, 256, 407–408, 487, 523–533, 535–536, 550, 560–571

model, 27–38

project

complexity, 150, 158

management, 150, 152–158

quality, 499–509

Stability, 149, 176, 184, 430, 450–452,

456–458, 468, 488, 565, 567, 569

Study program, 188

Substitution, 274, 378–379, 381–382, 386, 426

Success conditions, 44, 46–52

T

Teaching and learning, 92

Technology forecasting, 425–434

Trust, 50, 103–104, 108, 113, 116, 188, 233–234, 303, 334–336, 438–439, 446, 536–541, 543–544, 549

Turnover, 45–46, 113, 349, 384, 470

U

UML, 35, 69–70, 85, 89, 270–273, 275–276, 283–284, 286, 288–290, 307, 310–311, 315, 322, 411–413, 415, 417–419, 563

Uncertainty, 137, 150, 152, 226, 450–452, 454, 456–458

University–industry relations, 99, 108

Unknown demand, 378, 382

User

-centred software development, 499

-defined quality tool, 213–221

feedback, 411–422

testing, 177, 179, 183–184, 499–509

V

Value of knowledge, 204–205, 209, 238

Veil of ignorance, 100–101, 105, 108

Virtual enterprise (VE), 233–239

W

Web 2.0, 163–165, 169

Web

-based enterprise systems, 46

Web (*cont.*)

engineering, 295–304
portals, 175–178, 181, 183–184, 463
service, 3–4, 6–7, 13, 32–33, 216, 284–285,
320, 369–371, 374, 414–415

Weight determination, 478, 481

Wizard-of-Oz prototyping, 500–501, 503

Workflow, 20, 23, 28–29, 32–34, 36–37, 169,
177, 283–286, 288, 291, 322, 350,
414, 464, 500, 504, 533

X

XML

document, 31–32, 213–221, 289
nets, 28, 31–33, 35–38