

Proceedings of the

27th IEEE International Conference on Software Maintenance



September 25-30, 2011

Copyright and Reprint Permission: Abstracting is permitted with credit to the source. Libraries are permitted to photocopy beyond the limit of U.S. copyright law for private use of patrons those articles in this volume that carry a code at the bottom of the first page, provided the per-copy fee indicated in the code is paid through Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923. For other copying, reprint or republication permission, write to IEEE Copyrights Manager, IEEE Operations Center, 445 Hoes Lane, Piscataway, NJ 08854. All rights reserved. Copyright © 2011 by IEEE.

Editors

Andrian Marcus, Wayne State University, USA
James R. Cordy, Queen's University, Canada
Paolo Tonella, Fondazione Bruno Kessler, Italy

Sponsors

IEEE Computer Society
Technical Council on Software Engineering



Supporting Organizations

ABB Group
College of William & Mary
Software Improvement Group (SIG)
Wayne State University



Student Travel Support

The National Science Foundation (NSF), USA



Table of Contents

Front Matter

Foreword	viii
ICSM 2011 Organizing Committee	ix
ICSM 2011 Program Committee	x
<i>Additional Reviewers</i>	xii
Industry Track Program Committee	xiii
<i>Additional Reviewers</i>	xiii
Early Research Achievements Program Committee	xiv
<i>Additional Reviewers</i>	xiv
Tool Demonstrations Program Committee	xv
<i>Additional Reviewers</i>	xv

Technical Papers

Keynotes

How to Steal a Botnet and What Can Happen When You Do	1
<i>Richard A. Kemmerer</i>	
Useful Software Engineering Research - Leading a Double-Agent Life	2
<i>Lionel C. Briand</i>	

Research Track

Faults and Regression Testing

Fault Interaction and its Repercussions	3
<i>Nicholas Digiuseppe and James Jones</i>	
A Novel Approach to Regression Test Selection for J2EE Applications	13
<i>Sheng Huang, Zhong Jie Li, Jun Zhu, Yanghua Xiao and Wei Wang</i>	
Localizing Fault-Inducing Program Edits Based on Spectrum Information	23
<i>Lingming Zhang, Miryung Kim and Sarfraz Khurshid</i>	

Impact Analysis

Automated Change Impact Analysis for Agent Systems	33
<i>Hoa Khanh Dam and Aditya Ghose</i>	
ImpactScale: Quantifying Change Impact to Predict Faults in Large Software Systems	43
<i>Kenichi Kobayashi, Akihiko Matsuo, Katsuro Inoue, Yasuhiro Hayase, Manabu Kamimura and Toshiaki Yoshino</i>	
A Seismology-inspired Approach to Study Change Propagation	53
<i>Salima Hassaine, Ferdaous Boughanmi, Yann-Gaël Guéhéneuc, Sylvie Hamel and Giuliano Antoniol</i>	

Dynamic Analysis

SCOTCH: Test-to-Code Traceability using Slicing and Conceptual Coupling	63
<i>Abdallah Qusef, Gabriele Bavota, Rocco Oliveto, Andrea De Lucia and Dave Binkley</i>	
Identifying Distributed Features in SOA by Mining Dynamic Call Trees	73
<i>Anis Yousefi and Kamran Sartipi</i>	
Identifying Performance Deviations in Thread Pools	83
<i>Mark D. Syer, Bram Adams and Ahmed E. Hassan</i>	

Research Track

Natural Language Analysis

Mining Java Class Naming Conventions	93
<i>Simon Butler, Michel Wermelinger, Yijun Yu and Helen Sharp</i>	
Generating Natural Language Summaries for Crosscutting Source Code Concerns	103
<i>Sarah Rastkar, Gail C. Murphy and Alexander W. J. Bradley</i>	
Expanding Identifiers to Normalize Source Code Vocabulary	113
<i>Dawn Lawrie and Dave Binkley</i>	

Traceability

Using Tactic Traceability Information Models to Reduce the Risk of Architectural Degradation during System Maintenance	123
<i>Mehdi Mirakhorli and Jane Cleland-Huang</i>	
On Integrating Orthogonal Information Retrieval Methods to Improve Traceability Link Recovery	133
<i>Malcom Gethers, Rocco Oliveto, Denys Poshyvanyk and Andrea De Lucia</i>	
Structural Conformance Checking with Design Tests: An Evaluation of Usability and Scalability	143
<i>João Brunet, Dalton Dario Serey Guerrero and Jorge Figueiredo</i>	

Migration and Refactoring

MoMS: Multi-objective Miniaturization of Software	153
<i>Nasir Ali, Wei Wu, Giuliano Antoniol, Massimiliano Di Penta, Yann-Gaël Guéhéneuc and Jane Huffman Hayes</i>	
A Method Engineering Based Legacy to SOA Migration Method	163
<i>Ravi Khadka, Slinger Jansen, Amir M. Saeidi, Jurriaan Hage and Gijs Reijnders</i>	
Identifying Overly Strong Conditions in Refactoring Implementations	173
<i>Gustavo Soares, Melina Mongiovi and Rohit Gheyi</i>	

Components

Graph-based Detection of Library API Imitations	183
<i>Chengnian Sun, Siau-Cheng Khoo and Shao Jie Zhang</i>	
Crossing the Boundaries While Analyzing Heterogeneous Component-Based Software Systems	193
<i>Amir Reza Yazdanshenas and Leon Moonen</i>	
On the Maintenance of UI-integrated Mashup Applications	203
<i>Maxim Shevertalov and Spiros Mancoridis</i>	

Program Comprehension

An Exploratory Study of Feature Location Process: Distinct Phases, Recurring Patterns, and Elementary Actions	213
<i>Jinshui Wang, Xin Peng, Zhenchang Xing and Wenyun Zhao</i>	
Exploiting Text Mining Techniques in the Analysis of Execution Traces	223
<i>Heidar Pirzadeh, Abdelwahab Hamou-Lhadj and Mohak Shah</i>	
An Evaluation of the Strategies of Sorting, Filtering, and Grouping API Methods for Code Completion	233
<i>Daqing Hou and Dave Pletcher</i>	

Reliability and Quality

A Probabilistic Software Quality Model	243
<i>Tibor Bakota, Péter Hegedüs, Péter Körtvélyesi, Rudolf Ferenc and Tibor Gyimóthy</i>	
Predicting Post-release Defects Using Pre-release Field Testing Results	253
<i>Fouse Khomh, Brian Chan, Ying Zou, Anand Sinha and Dave Dietz</i>	
Sahara: Guiding the Debugging of Failed Software Upgrades	263
<i>Rekha Bachwani, Olivier Crameri, Ricardo Bianchini, Dejan Kostic and Willy Zwaenepoel</i>	

Research Track

Clones

- Late Propagation in Software Clones 273
Liliane Barbour, Foutse Khomh and Ying Zou
- File Cloning in Open Source Java Projects: The Good, The Bad, and The Ugly 283
Joel Ossher, Hitesh Sajjani and Cristina Lopes
- An Automatic Framework for Extracting and Classifying Near-Miss Clone Genealogies 293
Ripon K. Saha, Chanchal K. Roy and Kevin A. Schneider

Metrics

- Using Source Code Metrics to Predict Change-Prone Java Interfaces 303
Daniele Romano and Martin Pinzger
- You Can't Control the Unfamiliar: A Study on the Relations Between Aggregation Techniques for Software Metrics 313
Bogdan Vasilescu, Alexander Serebrenik and Mark Van Den Brand
- Understanding a Developer Social Network and its Evolution 323
Qiaona Hong, Sunghun Kim, Shing-Chi Cheung and Christian Bird

Clustering and Categorization

- Classifying Field Crash Reports for Fixing Bugs : A Case Study of Mozilla Firefox 333
Tejinder Dhaliwal, Foutse Khomh and Ying Zou
- Categorizing Software Applications for Maintenance 343
Collin Mcmillan, Mario Linares-Vásquez, Denys Poshyvanyk and Mark Grechanik
- Evaluating Software Clustering using Multiple Simulated Authoritative Decompositions 353
Mark Shtern and Vassilios Tzerpos

Industry Track

Testing

- Industrial Experiences with Automated Regression Testing of a Legacy Database Application 362
Erik Rogstad, Lionel Briand, Ronny Dalberg, Marianne Rynning and Erik Arisholm
- Regression Testing in Software as a Service: An Industrial Case Study 372
Hema Srikanth and Myra Cohen
- A Clustering Approach to Improving Test Case Prioritization: An Industrial Case Study 382
Ryan Carlson, Hyunsook Do and Anne Denton

Reverse Engineering

- Code Hot Spot: A Tool for Extraction and Analysis of Code Change History 392
Will Snipes, Brian Robinson and Emerson Murphy-Hill
- An Integration Resolution Algorithm for Mining Multiple Branches in Version Control Systems 402
Alexander Tarvo, Thomas Zimmermann and Jacek Czerwonka
- Relating Developers' Concepts and Artefact Vocabulary in a Financial Software Module 412
Tezcan Dilshener and Michel Wermelinger

Evolution and Migration

- Incremental and Iterative Reengineering towards Software Product Line: An Industrial Case Study 418
Gang Zhang, Liwei Shen, Xin Peng, Zhenchang Xing and Wenyun Zhao
- The Evolution of Information Systems. A Case Study on Document Management 428
Paolo Salvaneschi
- Testing & Quality Assurance in Data Migration Projects 438
Florian Matthes, Christopher Schulz and Klaus Haller

Industry Track

Program Analysis and Verification

- Precise Detection of Un-initialized Variables in Large, Real-life COBOL Programs in Presence of Unrealizable Paths 448
Rahul Jiresal, Adnan Contractor and Ravindra Naik
- Type-preserving Heap Profiler for C++ 457
József Mihalicza, Zoltán Porkoláb and Ábel Gábor
- Analyzing the Effects of Formal Methods on the Development of Industrial Control Software 467
Jan Friso Groote, Ammar Osaiweran and Jacco H. Wesselius

Metrics and Estimation

- Source Code Comprehension Strategies and Metrics to Predict Comprehension Effort in Software Maintenance and Evolution Tasks - An Empirical Study with Industry Practitioners 473
Kazuki Nishizono, Shuji Morisaki, Rodrigo Vivanco and Kenichi Matsumoto
- Estimating Software Maintenance Effort from Use Cases: An Industrial Case Study 482
Yan Ku, Jing Du, Ye Yang and Qing Wang

Early Research Achievements Track

Linguistic Analysis of Software Artifacts

- Toward a Metrics Suite for Source Code Lexicons 492
Lauren R. Biggers, Brian P. Eddy, Nicholas A. Kraft and Letha H. Etzkorn
- A Comparison of Stemmers on Source Code Identifiers for Software Search 496
Andrew Wiese, Valerie Ho and Emily Hill
- Clustering and Lexical Information Support for the Recovery of Design Pattern in Source Code 500
Simone Romano, Giuseppe Scanniello, Michele Risi and Carmine Gravino
- Code Convention Adherence in Evolving Software 504
Michael Smit, Barry Gergel, H. James Hoover and Eleni Stroulia
- Bug Report Quality Evaluation with Topic Modeling 508
Shusi Yu, Shuigeng Zhou and Jihong Guan

Software Changes and Maintainability

- SE² Model to Support Software Evolution 512
Huzefa Kagdi, Malcom Gethers and Denys Poshyvanyk
- Measuring Maintainability of Spreadsheets in the Wild 516
José Pedro Correia and Miguel Alexandre Ferreira
- Using Stereotypes to Help Characterize Commits 520
Natalia Dragan, Michael L. Collard, Maen Hammad and Jonathan I. Maletic
- Source Code Survival with the Kaplan Meier Estimator 524
Giuseppe Scanniello

Managing and Supporting Software Maintenance Activities

- Tracking Technical Debt: An Exploratory Case Study 528
Yuepu Guo, Carolyn Seaman, Rebeka Gomes, Antonio Cavalcanti, Graziela Tonin, Fabio Q. B. Da Silva, André L. M. Santos and Claurton Siebra
- Measuring Disruption from Software Evolution Activities Using Graph-Based Metrics 532
Prashant Paymal, Rajvardhan Patil, Sanjukta Bhowmick and Harvey Siy
- Clustering and Recommending Collections of Code Relevant to Tasks 536
Seonah Lee and Sungwon Kang
- Dependency Profiles for Software Architecture Evaluations 540
Eric Bouwers, Arie Van Deursen and Joost Visser

Post Doctoral Symposium Track

Empirical Assessment of UML Class Diagram Layouts Based on Architectural Importance <i>Bonita Sharif</i>	544
Emergent Laws of Method and Class Stereotypes in Object Oriented Software <i>Natalia Dragan</i>	550
A Logic Meta-Programming Foundation for Example-Driven Pattern Detection in Object-Oriented Programs <i>Coen De Roover</i>	556
Evidence-based Software Process Recovery: A Post-Doctoral View <i>Abram Hindle</i>	562
Sociotechnical Coordination and Collaboration in Open Source Software <i>Christian Bird</i>	568

Tool Demonstrations Track

A Tool for Combinatorial-based Prioritization and Reduction of User-Session-Based Test Suites <i>Sreedevi Sampath, Renee Bryce, Sachin Jain and Schuyler Manchester</i>	574
MARBLE. A Business Process Archeology Tool <i>Ricardo Pérez-Castillo, María Fernández-Ropero, Ignacio García-Rodríguez de Guzmán and Mario Piattini</i>	578
Program Querying with a SOUL: The Barista Tool Suite <i>Carlos Noguera, Coen De Roover, Andy Kellens and Viviane Jonckers</i>	582
WebDiff: A Generic Differencing Service for Software Artifacts <i>Nikolaos Tsantalis, Natalia Negara and Eleni Stroulia</i>	586
EQ: Checking the Implementation of Equality in Java <i>Chandan R. Rupakheti and Daqing Hou</i>	590
Maleku: An Evolutionary Visual Software Analytics Tool for Providing Insights into Software Evolution <i>Antonio González Torres, Roberto Therón, Francisco José García Peñalvo, Michel Wermelinger and Yijun Yu</i>	594

Tutorials

Effective Mining of Software Repositories <i>Marco D'Ambros and Romain Robbes</i>	598
Practical Combinatorial (t-way) Methods for Detecting Complex Faults in Regression Testing <i>Rick Kuhn and Raghu Kacker</i>	599
Research Methods in Computer Science <i>Serge Demeyer</i>	600

Foreword

Welcome to the 27th IEEE International Conference on Software Maintenance in Williamsburg, Virginia, USA. ICSM is continuing its journey around the world in a new location. The conference is hosted by the Williamsburg Lodge, situated in the heart of Historical Williamsburg. We hope you will enjoy the location as much as the conference.

ICSM 2011 is the result of a long effort undertaken by many people. The Organizing Committee includes 19 people whose work was also supported by seven student volunteers. The Program Committees for all tracks included 112 people and many additional reviewers contributed to the review process. The names of these volunteers are listed on the following pages and we want to thank all of them for their great work and contributions. ICSM would not exist without the effort of such people.

We want to thank the technical sponsors of the conference, the IEEE Computer Society and the IEEE Technical Council on Software Engineering, for their help and support. We extend our gratitude to our supporters for their generous contributions: The College of William & Mary, Wayne State University, ABB, and SIG.

Four additional events are collocated with ICSM this year: the 11th IEEE International Working Conference on Source Code Analysis and Manipulation (SCAM), the 13th IEEE International Symposium on Web Systems Evolution (WSE), the 6th IEEE International Workshop on Visualizing Software for Understanding and Analysis (VISSOFT), and the International Workshop on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems (MESOCA). In addition to the main Research Track, ICSM 2011 features three Tutorials, the Early Research Achievements (ERA) Track, the Industry Track, a Tool Demo Track, a Mid-career Doctoral Symposium, a Post-Doctoral Symposium, and two invited keynotes. This year brings a premiere, as ICSM 2011 will award and present the Most Influential Paper from ICSM 2001.

The Research Track includes 36 papers. These were selected from 127 submissions, submitted by 401 authors from 28 countries. Each paper was reviewed by at least three members of the Program Committee. The Program Committee had 77 members from 19 countries. Additional 89 external reviewers helped the PC with the reviews. The reviews were lively discussed online during two weeks and final decisions were made based on the reviews and discussions.

The ERA Track will feature 13 papers, which will also be presented informally in a poster session. The Tool Demo Track includes six tool demonstrations, which will be presented formally and also informally during the poster session. The Industrial Track includes 14 papers, while the Dissertation Session includes five presentations. Last but not least, the ICSM program includes three half-day tutorials and a half-day Mid-career doctoral symposium.

Two keynote presentations are scheduled this year. We are excited to have Dr. Richard Kemmerer and Dr. Lionel Briand giving the keynotes.

We hope you will have a great time and an unforgettable experience at ICSM 2011.

Andrian Marcus,
ICSM 2011 General Chair,
Wayne State University,
Detroit, USA

James R. Cordy,
ICSM 2011 Program Co-Chair
Queen's University, Kingston,
Canada

Paolo Tonella,
ICSM 2011 Program Co-Chair
Fondazione Bruno Kessler,
Trento, Italy

ICSM 2011 Organizing Committee

General Chair

Andrian Marcus, Wayne State University, USA

Program Chairs

James R. Cordy, Queen's University, Canada
Paolo Tonella, Fondazione Bruno Kessler, Italy

Industry Track Chairs

Mark Grechanik, Accenture Labs, USA
Joost Visser, Software Improvement Group, Netherlands

Early Research Achievements Track Chairs

Massimiliano Di Penta, University of Sannio, Italy
Miryung Kim, University of Texas at Austin, USA

Tool Demonstrations Track Chairs

Rocco Oliveto, University of Molise, Italy
Denys Poshyvanyk, College of William and Mary, USA

Doctoral Symposium Chairs

Serge Demeyer, University of Antwerp, Belgium
Mike Godfrey, University of Waterloo, Canada

Tutorials Track Chairs

Hausi A. Müller, University of Victoria, Canada
Lori Pollock, University of Delaware, USA

Finance Chair

Michael Collard, University of Akron, USA

Proceedings Chair

Huzefa Kagdi, Wichita State University, USA

Local Arrangements Chair

Denys Poshyvanyk, College of William and Mary, USA

Publicity Chairs

Sonia Haiduc, Wayne State University, USA
Romain Robbes, University of Chile, Chile

Student Volunteers Chair

Collin McMillan, College of William and Mary, USA

Web Chair

Scott Ohlemacher, Wayne State University, USA

ICSM 2011 Program Committee

Nicolas Anquetil, École des Mines de Nantes, France
Giuliano Antonioli, École Polytechnique de Montréal, Canada
Francoise Balmas, University Paris 8, France
Árpád Beszédes, University of Szeged, Hungary
Dirk Beyer, University of Passau, Germany
Dave Binkley, Loyola College, USA
Sue Black, University College London, UK
Shawn Bohner, Rose-Hulman Institute of Technology, USA
Jim Buckley, University of Limerick, Ireland
Mariano Ceccato, Fondazione Bruno Kessler, Italy
Michael Collard, University of Akron, USA
Andrea De Lucia, University of Salerno, Italy
Thomas Dean, Queen's University, Canada
Serge Demeyer, University of Antwerp, Belgium
Massimiliano Di Penta, University of Sannio, Italy
Stephane Ducasse, INRIA Lille, France
Rudolf Ferenc, University of Szeged, Hungary
Harald Gall, University of Zurich, Switzerland
Daniel German, University of Victoria, Canada
Tudor Girba, University of Bern, Switzerland
Mike Godfrey, University of Waterloo, Canada
Mark Grechanik, Accenture Labs, USA
Yann-Gaël Guéhéneuc, École Polytechnique de Montréal, Canada
Tibor Gyimothy, University of Szeged, Hungary
Mark Harman, University College London, UK
Ahmed E. Hassan, Queen's University, Canada
Abram Hindle, University of Waterloo, Canada
Mehdi Jazayeri, University of Lugano, Switzerland
Huzefa Kagdi, Wichita State University, USA
Andy Kellens, Vrije Universiteit Brussel, Belgium
Holger Kienle, University of Victoria, Canada
Miryung Kim, University of Texas at Austin, USA
Raghavan Komondoor, Indian Institute of Science, India
Bogdan Korel, Illinois Institute of Technology, USA
Rainer Koschke, University of Bremen, Germany
Jens Krinke, University College London, UK
Filippo Lanubile, University of Bari, Italy
Michele Lanza, University of Lugano, Switzerland
Zheng Li, University College London, UK
Panos Linos, Butler University, USA
Jonathan Maletic, Kent State University, USA
Spiros Mancoridis, Drexel University, USA
Radu Marinescu, Politehnica University of Timisoara, Romania
Tom Mens, Universit de Mons-Hainaut, Belgium
Ettore Merlo, École Polytechnique de Montréal, Canada
Ana Milanova, Rensselaer Polytechnic Institute, USA
Leon Moonen, Simula Research Laboratory, Norway
Gail Murphy, University of British Columbia, Canada
Oscar Nierstrasz, University of Bern, Switzerland
Rocco Oliveto, University of Molise, Italy
Denys Poshyvanyk, College of William and Mary, USA
Vaclav Rajlich, Wayne State University, USA
Filippo Ricca, University of Genova, Italy
Juergen Rilling, Concordia University, Canada

Gregorio Robles, Universidad Rey Juan Carlos, Spain
Gregg Rothermel, University of Nebraska, USA
Chanchal K. Roy, University of Saskatchewan, Canada
Kamran Sartipi, McMaster University, Canada
Giuseppe Scanniello, Università della Basilicata, Italy
Margaret-Anne Storey, University of Victoria, Canada
Eleni Stroulia, University of Alberta, Canada
Tarja Systä, Tampere University of Technology, Finland
Tetsuo Tamai, University of Tokyo, Japan
Eli Televich, Virginia Tech, USA
Vassilios Tzerpos, York University, Canada
Arie van Deursen, Delft University of Technology, Netherlands
Joost Visser, Software Improvement Group, Netherlands
Claudia Werner, Federal University of Rio de Janeiro, Brazil
Norman Wilde, University of West Florida, USA
Kenny Wong, University of Alberta, Canada
Carl Worms, Credit Suisse, Switzerland
Tao Xie, North Carolina State University, USA
Yunwen Ye, Software Research Associates Inc., Japan
Andreas Zeller, Saarland University, Germany
Lu Zhang, Peking University, China
Thomas Zimmermann, Microsoft Corporation, USA
Ying Zou, Queen's University, Canada

ICSM 2011 Additional Reviewers

Abdulkareem Alali
Venera Arnaoudova
Jean Baptiste Arnaud
Tibor Bakota
Liliane Barbour
Brian Bartman
Gabriele Bavota
Olga Baysal
Muhammad Usman Bhatti
Walter Binder
Eric Bouwers
Fabio Calefato
Gwenael Casaccio
José Pedro Correia
Marco D'Ambros
Yingnong Dang
Julius Davis
Frederik De Bleser
Sylvain Degrandart
Tejinder Dhaliwal
Bogdan Dit
Leandro Doctors
Natalia Dragan
Michael English
Fausto Fasano
Miguel Alexandre Ferreira
Lajos Fülöp
Francois Gauthier
Tamás Gergely
Malcom Gethers
Mathieu Goeminne
Lars Grammel
Carmine Gravino
Latifa Guerrouj
Imed Hammouda
Dan Hao
Wei He
Péter Hegedüs
Gabriella Kakuja-Tóth
Foutse Khomh
Segla Kpodjedo
Péter Körtvélyesi
Ahmed Lamkanfi
Jannik Laval
Thierry Lavoie
Maurizio Leotta
Stefan Loewe
Andrea Magdaleno
Cristina Marinescu
Collin Mcmillan
Na Meng
Evan Moritz
Del Myers
Csaba Nagy
Yitao Ni
Carlos Noguera
Fabrizio Perin
Steven Raemaekers
Saidur Rahman
Jorge Ressia
Michele Risi
Ripon Saha
Rodrigo Santos
Marcelo Schots
Lajos Schrettner
Xander Schrijen
Niko Schwarz
Kiran Shakya
Emad Shihab
Mark Shtern
István Siket
Quinten David Soetens
Neset Sozen
Andrew Sutton
Kunal Taneja
Christoph Treude
Adrian Trifu
Radu Vanciu
Toon Verwaest
László Vidács
Philipp Wendler
Erwann Wernli
Leon Wilson
Zhihong Xu
Tingting Yu
Hao Yuan
Linghao Zhang
Hao Zhong
Minhaz Zibran

Industry Track Program Committee

Robert Baggen, TÜV Informations Technik, Germany
Magiel Bruntink, Software Improvement Group, The Netherlands
Christoph Csallner, University of Texas at Arlington, USA
Chen Fu, Accenture Technology Labs
James Jones, University of California, Irvine, USA
Carola Lilienthal, C1 WPS, Germany
Jonathan Maletic, Kent State University, USA
Atif Memon, University of Maryland, USA
Henry Muccini, University of L'Aquila, Italy
Ipek Ozkaya, Software Engineering Institute, USA
Martin Pinzger, Technical University Delft, The Netherlands
Denys Poshyvanyk, College of William and Mary, USA
Brian Robinson, ABB Research, USA
Christa Schwanninger, Siemens
Mike Smit, University of Alberta, Canada
Eleni Stroulia, University of Alberta, Canada
James Terwilliger, Microsoft, USA
Eli Tilevich, Virginia Tech, USA
Carl Worms, Credit Suisse, Switzerland
Yunwen Ye, Software Research Associates, Japan

Industry Track Additional Reviewers

Shabnam Aboughadareh
Mithun Acharya
Jeroen Arnoldus
Vinay Augustine
Amine Chigani
Bogdan Dit
Alexandre Ferreira
Malcom Gethers
James Hoover
Ishtiaque Hussain
Mainul Islam
Sam Klock
Collin Mcmillan
José Pedro Correia Miguel
Tuan Nguyen
Tuli Nivas
Ariadi Nugroho
Hari Pyla
Daniele Romano
Sarker Ahmed Rume

Early Research Achievements Program Committee

Adrian Kuhn, University of Bern, Switzerland
Anthony Cleve, University of Namur, Belgium
Bram Adams, SAIL, Queen's University, Canada
Chen Fu, Accenture Technology Labs, USA
Christian Bird, Microsoft Research, USA
Christoph Csallner, University of Texas at Arlington, USA
David Lo, Singapore Management University, Singapore
Gerardo Canfora, University of Sannio, Italy
Huzefa Kagdi, Wichita State University, USA
Iulian Neamtiu, University of California, Riverside, USA
Marco D'Ambros, University of Lugano
Martin Pinzger, TU Delft, The Netherlands
Massimiliano Di Penta, University of Sannio, Italy
Miryung Kim, University of Texas, Austin, USA
Naouel Moha, Université du Québec à Montréal, Canada
Nicholas A. Kraft, The University of Alabama, USA
Rahul Premraj, Vrije University Amsterdam, Netherlands
Reid Holmes, University of Waterloo, Canada
Romain Robbes, University of Chile, Chile
Saurabh Sinha, IBM Research, India
Yuanfang Cai, Drexel University, USA
Zhenchang Xing, National University of Singapore, USA

Early Research Achievements Track Additional Reviewers

Shabnam Aboughadareh
Sarker Ahmed
Ishtiaque Hussain
Mainul Islam
Meiyappan Nagappan
Daniele Romano
Rumee Yinxing Xue

Tool Demonstrations Program Committee

Mario Luca Bernardi, University of Sannio, Benevento, Italy
Anthony Cleve, University of Namur & Université Libre de Bruxelles, Belgium
Christoph Csallner, University of Texas, Arlington, USA
Marco D'Ambros, University of Lugano, Switzerland
Fausto Fasano, University of Molise, Italy
Foutse Khomh, Queens University, Canada
Mircea Lungu, University of Berne, Switzerland
Collin McMillan, College of William and Mary, USA
Rocco Oliveto, University of Molise, Italy
Denys Poshyvanyk, College of William and Mary, USA
Romain Robbes, University of Chile, Chile

Tool Demonstrations Track Additional Reviewers

Shabnam Aboughadareh
Ishtiaque Hussain
Mainul Islam
Sarker Ahmed Rume

MARBLE. A Business Process Archeology Tool

Ricardo Pérez-Castillo, María Fernández-Ropero, Ignacio García-Rodríguez de Guzmán and Mario Piattini

Alarcos Research Group, University of Castilla-La Mancha

Paseo de la Universidad 4 13071, Ciudad Real, Spain

{ricardo.pdelcastillo, marias.fernandez, ignacio.grodriguez, mario.piattini}@uclm.es

Abstract—Modernization of legacy information systems is usually triggered by the need of introducing new business requirements or due to the technology obsolescence. During modernization software projects, there exists a lot of business knowledge that was embedded in source code owing to progressive maintenance, which is not present anywhere else. In order to preserve embedded business knowledge this paper presents MARBLE, a tool to recover business processes from legacy information systems. MARBLE provides an extensible framework, thus it is developed as an Eclipse™ plugin to ensure its future extension. So far, MARBLE supports Java-based system and focuses mainly on legacy source code. To facilitate its adoption in the industry, MARBLE has been applied in some real-life modernization projects, which demonstrated that MARBLE is suitable to retrieve business processes and facilitated its continuous improvement to reach an appropriate maturity level.

Keywords—Reverse Engineering; Business Process; Model-Driven Development; Knowledge Discovery Metamodel

I. INTRODUCTION

Reverse engineering techniques have become very important within the maintenance process providing several benefits. Firstly, reverse engineering allows maintainers to retrieve abstract representations to facilitate the comprehension of different legacy systems. For example, it focuses on relational databases [6], aspect oriented systems [3], quality of the system design [8], links between e-mail and source code [2], and so on. Secondly, abstract representations obtained by reverse engineering from legacy systems can be refactored to improve their maintainability or add new functionalities to evolve legacy systems.

To address the mentioned maintenance activities, reverse engineering techniques are nowadays well-supported by tools which often obtain artifacts at system design abstraction level (e.g., class or sequence diagrams from source code) [5]. However, software engineering industry is demanding additional reverse engineering techniques and tools to retrieve business-aware artifacts at higher abstraction level [11]. In fact, the Software Engineering Institute (SEI) argues that business rules recovery is the cornerstone to evolutionary maintenance towards modern paradigms like Service Oriented Architecture (SOA) [12].

To meet these demands, business process archeology has emerged as a set of techniques and tools to recover business processes from source code [15]. Maintenance benefits of business process archeology are that they preserve business behavior buried in legacy source code and it retrieves business

processes providing more opportunities for refactoring (due to the higher abstraction level).

There are some techniques in literature that support business process recovery. For example *Zou et al.* [19] recover workflows by statically analyzing source code and applying some heuristic rules to discover business knowledge. *Paradauskas et al.* [14] retrieve business knowledge through the inspection of the data stored in databases. These studies rely solely on static analysis. Thereby, other solutions have been suggested based on dynamic analysis. For example, *Di Francescomarino et al.* [7] consider graphical user interfaces of web applications to discover business processes. *Cai et al.* [4] combine requirement reacquisition based on use cases with dynamic and static analysis techniques. Finally, *Van der Aalst et al.* [18] focus on mining business processes from event logs registered during system execution.

This paper particularly presents MARBLE™, a technique and tool, supporting business process archeology [15]. MARBLE™ is an extensible framework, although it so far supports static analysis of legacy source code in a similar way than technique proposed by *Zou et al.* However, MARBLE™ applies a set of business patterns formalized by model transformations. The main difference regarding mentioned proposals is that MARBLE™ is easily automatable and highly extensible due to it follows model-driven architecture principles (i.e., it considers all involved software artifacts as models, and it provides formal transformations to move models between different abstraction levels).

MARBLE™ tool [1] is implemented as an *Eclipse*™ plugin improving its extension and integration with other techniques and tools as well as to facilitate its adoption by the industry. The tool creates and manages an entire repository by integrating code models from legacy information systems. After that, the tool allows maintainers to discover, visualize and edit business process models.

The effectiveness and suitability of the tool has been demonstrated through several industrial case studies which the tool has been applied to a healthcare system, an e-government system, enterprise systems, among other kinds of legacy systems.

The remaining of the paper is organized as follows: Section II explains MARBLE™ technique supported by the tool. Section III presents in detail functionalities and implementation details of MARBLE™ tool. Section IV briefly summarizes the case studies which the tool has been used. Finally, Section IV discusses the conclusions of this work.

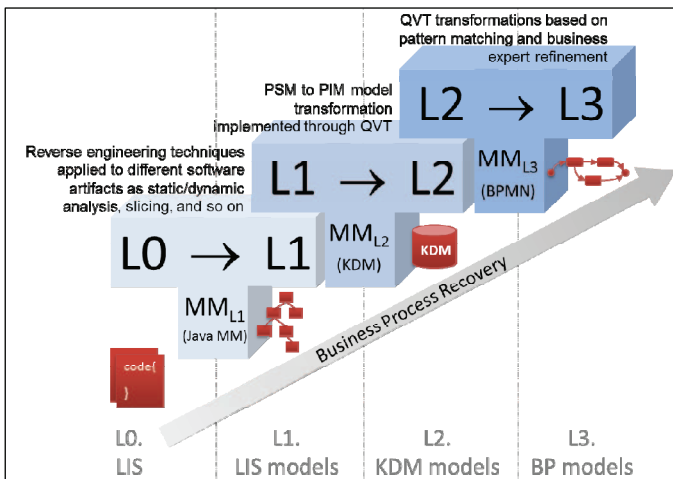


Figure 1. MARBLE™, the technique to support business process archeology

II. BUSINESS PROCESS ARCHAEOLOGY PROCEDURE

Business process archeology [15] studies the business processes in an organization by analyzing the existing software artifacts. The objective is to discover the business forces that motivated the construction of the enterprise information systems. On the one hand, traditional archeologists investigate several artifacts and situations, trying to understand what they are looking at, i.e., they must understand the cultural and civilizing forces that produced those artifacts. Similarly, a business process archeologist analyzes different legacy artifacts such as source code, databases and user interfaces and then tries to learn what the organization was thinking to understand why the organization developed the information system in a particular way. Business process archeology initiative is being progressively supported by new reverse engineering techniques and tools to retrieve and elicit the embedded business knowledge. Particularly, this paper focuses on MARBLE™ (*Modernization Approach for Recovering Business processes from Legacy Systems*) [15]. MARBLE™ is an extensible framework based on the ADM (Architecture-Driven Modernization) initiative proposed by the OMG (Object Management Group). ADM advocates carrying out reengineering processes by considering model-driven development principles.

In addition, MARBLE™ supports the KDM (Knowledge Discovery Metamodel) [10] standard proposed by the ADM initiative. KDM enables the representation and management of knowledge extracted by means of reverse engineering from all the different software artifacts of legacy systems in an integrated way. Thus, that legacy knowledge is gradually transformed into business processes. For this purpose, MARBLE™ is divided into four abstraction levels with three transformations among them (see Figure 1):

Level L0. This level represents the legacy information system in the real world, and is the source system to recover underlying business processes.

Level L1. This level represents several specific models, i.e., one model for each different software artifact involved in the archeology process like source code, database, user interface, and so on. Traditional reverse engineering techniques

[5] such as static analysis, dynamic analysis, program slicing, formal concept analysis, and so on, could be used to extract the knowledge from any software artifact and build PSM (Platform-Specific Model) models related to it. These PSM models are represented according to specific metamodels. For example, a Java metamodel may be used to model the legacy source code, or an SQL metamodel to represent the database schema, etc.

Level L2. This level consists of a single PIM (Platform-Independent Model) that represents the integrated view of the set of PSM models at L1. The KDM metamodel is used so that L2 works as a KDM repository that can be progressively populated with knowledge extracted from the different legacy artifacts and information systems of an organization. In addition, L2 is represented in a technological-independent way due to the fact that KDM standard abstract all those details concerning the technological viewpoint (e.g. the program language). The transformation between levels L1 and L2 consists of a set of model transformations implemented using QVT (Query/View/Transformation).

Level L3. Finally, this level depicts, at the end of the archeology process, the business process models retrieved from a legacy system. Business process models at L3 represent a CIM (Computational-Independent Model) and are represented according to the BPMN (Business Process Modeling and Notation). This level closes the conceptual gap between the software architecture views and underlying business rules. The last transformation is based on a set of patterns. When a specific structure is detected in the KDM model at L2, each pattern indicates what elements should be built and how they are interrelated in the business process model at L3 [16]. This pattern matching is implemented through a QVT transformation [17].

The obtained models are a first sketch of the business process, which can be refined by business experts. This is due, for instance, to the fact that not all parts of current business processes are executed by legacy information systems, i.e., there are some manual business activities. Although experts post-intervention can be necessary, the first version of business processes, compared with business process redesign by business experts from scratch, represents a more efficient and less error-prone solution to get business process archeology. In addition, the business process redesign by experts from scratch might discard meaningful business knowledge that is only embedded in legacy information systems.

III. IMPLEMENTATION DETAILS

MARBLE™ technique is supported by a tool with the same name. MARBLE™ tool supports entirely the three model transformations presented between the four MARBLE™ levels. In addition, MARBLE™ allows maintainers and business experts to visualize and modify the first sketch of business processes retrieved after the last transformation.

MARBLE™ tool was initially developed as a desktop application. However, due to the extensible nature of MARBLE™, it is now online available as an *Eclipse™* plugin [1]. This fact ensures, in the future, easy extension and integration with other maintenance tools.

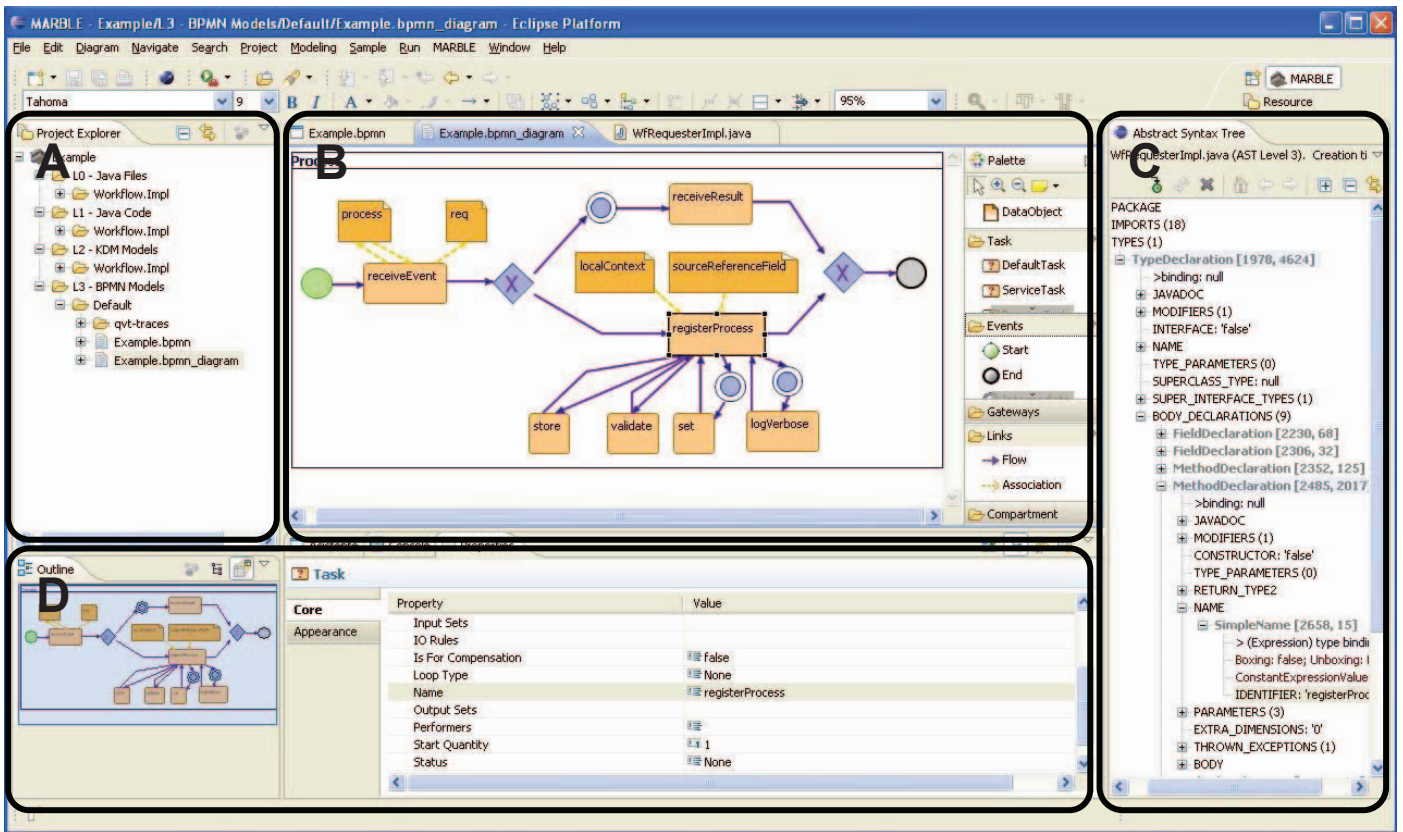


Figure 2. Screenshot of MARBLE™ tool with modules highlighted.

A. Technologies involved

Due to the fact that MARBLE™ technique provides a generic and extensible framework, MARBLE™ tool is firstly developed to support Java-based legacy systems. The tool is implemented through three key technologies. The first technology is *JavaCC*, which is a parser and scanner generator for Java. It is used to develop a Java static analyzer in the first transformation to obtain code models. Parsers for other programming languages may be developed in the future using *JavaCC* technology.

The second technology consists of two related *Eclipse™* frameworks: EMF (Eclipse Modeling Framework) and GMF (Graphical Modeling Framework). EMF is a modeling framework and code generation facility for building tools and other applications based on ECORE-compliant models. *ECORE* is a meta-metamodel, i.e. it is the metamodel proposed by the *Eclipse™* platform to define metamodels. EMF provides tools to automatically produce, from an ECORE-compliant metamodel, the source code of an editor to enable viewing and editing of the respective models. Moreover, GMF is used together with EMF to generate graphical editors from ECORE-compliant metamodels.

Finally, the third technology is QVT, which is a language specially developed to formalize model transformations. Model transformations involved in MARBLE™ tool are particularly implemented using QVTr (QVT Relations, the declarative language part of the QVT). In addition, model transformations are executed through *MediniQVT™*, an open source QVTr

transformation engine [9]. This transformation engine is integrated within the MARBLE™ plugin.

B. Main Functionalities

MARBLE tool functionalities are organized into four main modules. The first module is the project explorer (see Figure 2 A) which groups the models within each MARBLE level. Each model transformation can be done by selecting a whole level or a particular set of models.

The second model is the editor frame (see Figure 2 B) which visualizes the different kinds of models (i.e., java files, code models, KDM models and BPMN models). For example, the screenshot shows a business process model in the graphical editor, which additionally provides a palette of elements to facilitate the graphical edition of the model.

The third module provides a special tree editor with which visualize code models at L1 obtained by the *Java* parser developed (see Figure 2 C). This editor additionally enable feature location since it links code model elements of the abstract syntax tree with the java source code (i.e., an element of the code model can be located by clicking on a certain line of source code).

Finally, the fourth module provides additional relevant information about the particular model opened in the second module (see Figure 2 D). This module particularly provides an outline view as well as a property set view.

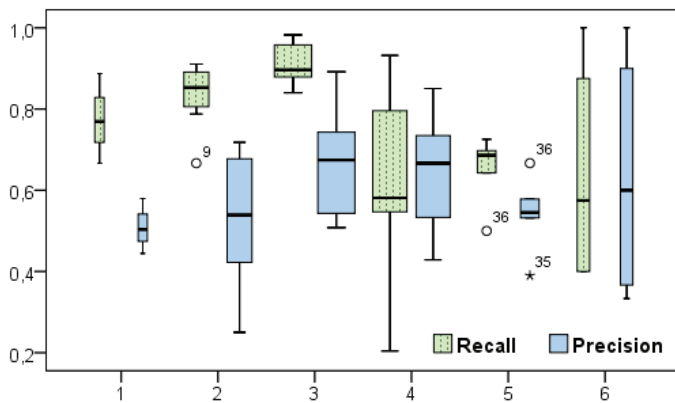


Figure 3. Effectiveness summary of case studies

IV. CASE STUDIES

MARBLE has been applied to several industrial case studies to recover business processes from a wide variety of legacy information systems. The conduction of these industrial case studies has allowed improving the tool and refining the MARBLE technique. So far, MARBLE has been used with six legacy systems in total: (i) a system managing a Spanish author organization; (ii) an open source CRM (Customer Relationship Management) system; (iii) an enterprise information system of the water and waste industry; (iv) an e-government system used in a Spanish local e-administration; (v) a high school LMS (Learning Management System); and finally (vi) an oncological evaluation system used in Austrian hospitals.

These studies evaluate the effectiveness and efficiency of the MARBLE technique applied through the tool. On one hand, effectiveness is measured through precision and recall. Precision measures the exactness or fidelity of the business processes recovered, whereas recall measures their completeness. These measures are computed regarding retrieved tasks and other related business process elements such as sequence flows, data objects and gateways. On the other hand, efficiency is evaluated through the time spent on the recovery as well as the MARBLE scalability to larger legacy information systems. Figure 3 summarizes results obtained from case studies regarding effectiveness. Precision and recall values vary from a system to another, although the value trend is a recall higher than precision. This means that MARBLE retrieves a great number of business activities although a few of them could be erroneous. In addition, a little set of activities could not be retrieved. Anyway, the results are appropriate regarding benchmark values around 0.5 [13].

V. CONCLUSIONS

MARBLE is a tool to retrieve business processes from legacy source code. MARBLE provide an extensible framework and it is therefore implemented as an Eclipse™ plugin. So far, MARBLE provides a *Java* parser to obtain code model, which are transformed and integrated in a model repository according to the KDM standard. After that, KDM model are transformed to business process models by applying business pattern recognition. MARBLE has already been applied to real-life modernization projects where business

process recovery was a mandatory requirement to preserve the embedded business knowledge.

ACKNOWLEDGMENT

This work was supported by the Spanish FPU Programme, and the R&D projects ALTAMIRA (JCCM, PII2I09-0106-2463), MOTERO (JCCM and FEDER, PEI11-0366-9449) and PEGASO/MAGO (TIN2009-13718-C02-01).

REFERENCES

- [1] Alarcos Research Group. Business Process Archeology. 2011 [cited 2011 10-06-2011]; Available from: <http://www.businessprocessarcheology.org/>.
- [2] Bacchelli, A., et al., Benchmarking Lightweight Techniques to Link E-Mails and Source Code, in WCRE'09. 2009. p. 205-214.
- [3] Bernardi, M.L., Reverse engineering of Aspect Oriented Systems to support their comprehension, evolution, testing and assessment, in CSMR'08. 2008, IEEE: Athens, Greece. p. 290-293.
- [4] Cai, Z., X. Yang, and W. Wang. Business Process Recovery for System Maintenance - An Empirical Approach, in ICSM'09. 2009, IEEE Computer Society. p. 399-402.
- [5] Canfora, G. and M. Di Penta. New Frontiers of Reverse Engineering. in Future of Software Engineering (FOSE'07). 2007: IEEE C. S.
- [6] Cleve, A. and J.L. Hainaut. Dynamic Analysis of SQL Statements for Data-Intensive Applications Reverse Engineering. in WCRE'08. 2008.
- [7] Di Francescomarino, C., A. Marchetto, and P. Tonella, Reverse Engineering of Business Processes exposed as Web Applications, in CSMR'09. 2009, IEEE C. S. p. 139-148.
- [8] Foutse, K., SQUAD: Software Quality Understanding through the Analysis of Design, in WCRE'09. 2009. p. 303-306.
- [9] ikv++, Medini QVT. 2008, ikv++ technologies ag.
- [10] ISO/IEC, ISO/IEC DIS 19506. Knowledge Discovery Meta-model (KDM), v1.1 (Architecture-Driven Modernization). 2009, ISO/IEC. p. 302.
- [11] Khusidman, V. and W. Ulrich, Architecture-Driven Modernization: Transforming the Enterprise. 2007, OMG. p. 7.
- [12] Lewis, G.A., D.B. Smith, and K. Kontogiannis, A Research Agenda for Service-Oriented Architecture (SOA): Maintenance and Evolution of Service-Oriented Systems. 2010, Software Engineering Institute. p. 40.
- [13] Lucrédio, D., R. de M. Fortes, and J. Whittle, MOOGLE: A Model Search Engine, in Model Driven Engineering Languages and Systems, K. Czarniecki, et al., Editors. 2008, Springer. p. 296-310.
- [14] Paradauskas, B. and A. Laurikaitis, Business Knowledge Extraction from Legacy Information Systems. Journal of Information Technology and Control, 2006. 35(3): p. 214-221.
- [15] Pérez-Castillo, R., I.G.-R. de Guzmán, and M. Piattini, Business Process Archeology using MARBLE. Information and Software Technology, 2011. 53: p. 1023-1044.
- [16] Pérez-Castillo, R., et al., Business Process Patterns for Software Archeology, in 25th Annual ACM Symposium on Applied Computing (SAC'10). 2010, ACM: Sierre, Switzerland. p. 165-166.
- [17] Pérez-Castillo, R., I. García-Rodríguez de Guzmán, and M. Piattini, Implementing Business Process Recovery Patterns through QVT Transformations, in International Conference on Model Transformation (ICMT'10). 2010, Springer-Verlag: Málaga, Spain. p. 168-183.
- [18] Van der Aalst, W.M.P., et al., ProM : the process mining toolkit, in 7th International Conference on Business Process Management (BPM'09) - Demonstration Track. 2009, Springer-Verlag: Ulm, Germany. p. 1-4.
- [19] Zou, Y. and M. Hung, An Approach for Extracting Workflows from E-Commerce Applications, in 14th International Conference on Program Comprehension. 2006, IEEE Computer Society. p. 127-136.

Author Index

- Adams, Bram, 83
Ali, Nasir, 153
Antoniol, Giuliano, 53, 153
Arisholm, Erik, 362
Bachwani, Rekha, 263
Bakota, Tibor, 243
Barbour, Liliane, 273
Bavota, Gabriele, 63
Bhowmick, Sanjukta, 532
Bianchini, Ricardo, 263
Biggers, Lauren R., 492
Binkley, Dave, 63, 113
Bird, Christian, 323, 568
Boughanmi, Ferdaous, 53
Bouwers, Eric, 540
Bradley, Alexander W. J., 103
Brand, Mark Van Den, 313
Briand, Lionel C., 2, 362
Brunet, João, 143
Bryce, Renee, 574
Butler, Simon, 93
Carlson, Ryan, 382
Cavalcanti, Antonio, 528
Chan, Brian, 253
Cheung, Shing-Chi, 323
Cleland-Huang, Jane, 123
Cohen, Myra, 372
Collard, Michael L., 520
Contractor, Adnan, 448
Correia, José Pedro, 516
Crameri, Olivier, 263
Czerwonka, Jacek, 402
D'Ambros, Marco, 598
Dalberg, Ronny, 362
Dam, Hoa Khanh, 33
Demeyer, Serge, 600
Denton, Anne, 382
Deursen, Arie Van, 540
Dhaliwal, Tejinder, 333
Di Penta, Massimiliano, 153
Dietz, Dave, 253
Digiuseppe, Nicholas, 3
Dilshener, Tezcan, 412
Do, Hyunsook, 382
Dragan, Natalia, 520, 550
Du, Jing, 482
Eddy, Brian P., 492
Etzkorn, Letha H., 492
Ferenc, Rudolf, 243
Fernández-Roperro, María, 578
Ferreira, Miguel Alexandre, 516
Figueiredo, Jorge, 143
Gábor, Ábel, 457
Gergel, Barry, 504
Gethers, Malcom, 133, 512
Gheyi, Rohit, 173
Ghose, Aditya, 33
Gomes, Rebeka, 528
Gravino, Carmine, 500
Grechanik, Mark, 343
Groote, Jan Friso, 467
Guan, Jihong, 508
Guéhéneuc, Yann-Gaël, 53, 153
Guerrero, Dalton Dario Serey, 143
Guo, Yuepu, 528
Guzmán, Ignacio García-Rodríguez de, 578
Gyimóthy, Tibor, 243
Hage, Jurriaan, 163
Haller, Klaus, 438
Hamel, Sylvie, 53
Hammad, Maen, 520
Hamou-Lhadj, Abdelwahab, 223
Hassaine, Salima, 53
Hassan, Ahmed E., 83
Hayase, Yasuhiro, 43
Hayes, Jane Huffman, 153
Hegedüs, Péter, 243
Hill, Emily, 496
Hindle, Abram, 562
Ho, Valerie, 496
Hong, Qiaona, 323
Hoover, H. James, 504
Hou, Daqing, 233, 590
Huang, Sheng, 13
Inoue, Katsuro, 43
Jain, Sachin, 574

Jansen, Slinger, 163
 Jiresal, Rahul, 448
 Jonckers, Viviane, 582
 Jones, James, 3
 Kacker, Raghu, 599
 Kagdi, Huzefa, 512
 Kamimura, Manabu, 43
 Kang, Sungwon, 536
 Kellens, Andy, 582
 Kemmerer, Richard A., 1
 Khadka, Ravi, 163
 Khomh, Foutse, 253, 273
 Khoo, Siau-Cheng, 183
 Khurshid, Sarfraz, 23
 Kim, Miryung, 23
 Kim, Sunghun, 323
 Kobayashi, Kenichi, 43
 Komah, Foutse, 333
 Körtvélyesi, Péter, 243
 Kostic, Dejan, 263
 Kraft, Nicholas A., 492
 Ku, Yan, 482
 Kuhn, Rick, 599
 Lawrie, Dawn, 113
 Lee, Seonah, 536
 Li, Zhong Jie, 13
 Linares-Vásquez, Mario, 343
 Lopes, Cristina, 283
 Lucia, Andrea De, 133
 Lucia, Andrea De, 63
 Maletic, Jonathan I., 520
 Manchester, Schuyler, 574
 Mancoridis, Spiros, 203
 Matsumoto, Kenichi, 473
 Matsuo, Akihiko, 43
 Matthes, Florian, 438
 Mcmillan, Collin, 343
 Mihalicza, József, 457
 Mirakhorli, Mehdi, 123
 Mongiovi, Melina, 173
 Moonen, Leon, 193
 Morisaki, Shuji, 473
 Murphy-Hill, Emerson, 392
 Murphy, Gail C., 103
 Naik, Ravindra, 448
 Negara, Natalia, 586
 Nishizono, Kazuki, 473
 Noguera, Carlos, 582
 Oliveto, Rocco, 63, 133
 Osaiweran, Ammar, 467
 Ossher, Joel, 283
 Patil, Rajvardhan, 532
 Paymal, Prashant, 532
 Peñalvo, Francisco José García, 594
 Peng, Xin, 213, 418
 Pérez-Castillo, Ricardo, 578
 Piattini, Mario, 578
 Pinzger, Martin, 303
 Pirzadeh, Heidar, 223
 Pletcher, Dave, 233
 Porkoláb, Zoltán, 457
 Poshyvanyk, Denys, 133, 343, 512
 Qusef, Abdallah, 63
 Rastkar, Sarah, 103
 Reijnders, Gijs, 163
 Risi, Michele, 500
 Robbes, Romain, 598
 Robinson, Brian, 392
 Rogstad, Erik, 362
 Romano, Daniele, 303
 Romano, Simone, 500
 Roover, Coen De, 556, 582
 Roy, Chanchal K., 293
 Rupakheti, Chandan R., 590
 Rynning, Marianne, 362
 Saeidi, Amir M., 163
 Saha, Ripon K., 293
 Sajnani, Hitesh, 283
 Salvaneschi, Paolo, 428
 Sampath, Sreedevi, 574
 Santos, André L. M., 528
 Sartipi, Kamran, 73
 Scanniello, Giuseppe, 500, 524
 Schneider, Kevin A., 293
 Schulz, Christopher, 438
 Seaman, Carolyn, 528
 Serebrenik, Alexander, 313
 Shah, Mohak, 223
 Sharif, Bonita, 544
 Sharp, Helen, 93
 Shen, Liwei, 418
 Shevertalov, Maxim, 203

Shtern, Mark, 343
Siebra, Claurton, 528
Silva, Fabio Q. B. Da, 528
Sinha, Anand, 253
Siy, Harvey, 532
Smit, Michael, 504
Snipes, Will, 392
Soares, Gustavo, 173
Srikanth, Hema, 372
Stroulia, Eleni, 504, 586
Sun, Chengnian, 183
Syer, Mark D., 83
Tarvo, Alexander, 402
Therón, Roberto, 594
Tonin, Graziela, 528
Torres, Antonio González, 594
Tsantalis, Nikolaos, 586
Tzerpos, Vassilios, 343
Vasilescu, Bogdan, 313
Visser, Joost, 540
Vivanco, Rodrigo, 473
Wang, Jinshui, 213
Wang, Qing, 482
Wang, Wei, 13
Wermelinger, Michel, 93, 412, 594
Wesselius, Jacco H., 467
Wiese, Andrew, 496
Wu, Wei, 153
Xiao, Yanghua, 13
Xing, Zhenchang, 213, 418
Yang, Ye, 482
Yazdanshenas, Amir Reza, 193
Yoshino, Toshiaki, 43
Yousefi, Anis, 73
Yu, Shusi, 508
Yu, Yijun, 93, 594
Zhang, Gang, 418
Zhang, Lingming, 23
Zhang, Shao Jie, 183
Zhao, Wenyun, 213, 418
Zhou, Shuigeng, 508
Zhu, Jun, 13
Zimmermann, Thomas, 402
Zou, Ying, 253, 273, 333
Zwaenepoel, Willy, 263