

2011 18th Working Conference on Reverse Engineering

(WCRE 2011)

**Limerick, Ireland
17-20 October 2011**



IEEE Catalog Number: CFP11090-PRT
ISBN: 978-1-4577-1948-6

2011 18th Working Conference on Reverse Engineering

WCRE 2011

Table of Contents

Message from the General Chair.....	x
Message from the Program Chairs.....	xi
Organizing Committee.....	xiii
Steering Committee.....	xiv
Program Committee.....	xv
Additional Reviewers.....	xvii
Keynote I: Evolving Critical Systems.....	xviii
Keynote II: Architecture-Driven Modernization of the European Air Traffic Management System.....	xix
WCRE 2001 Most Influential Paper: Robust Parsing using Island Grammars Revisited.....	xxvi

Session I – Clones

Incremental Code Clone Detection: A PDG-based Approach	3
<i>Yoshiki Higo, Ueda Yasushi, Minoru Nishino, and Shinji Kusumoto</i>	
On the Effectiveness of Simhash for Detecting Near-Miss Clones in Large Scale Software Systems	13
<i>Md. Sharif Uddin, Chanchal K. Roy, Kevin A. Schneider, and Abram Hindle</i>	
Internet-scale Real-time Code Clone Search Via Multi-level Indexing	23
<i>Iman Keivanloo, Juergen Rilling, and Philippe Charland</i>	
An Investigation into the Impact of Software Licenses on Copy-and-paste Reuse among OSS Projects	28
<i>Yu Kashima, Yasuhiro Hayase, Norihiro Yoshida, Yuki Manabe, and Katsuro Inoue</i>	

Session II – Comprehension I

Automatic Segmentation of Method Code into Meaningful Blocks to Improve Readability	35
<i>Xiaoran Wang, Lori Pollock, and K. Vijay-Shanker</i>	
Requirements Traceability for Object Oriented Systems by Partitioning Source Code	45
<i>Nasir Ali, Yann-Gaël Guéhéneuc, and Giuliano Antoniol</i>	

Can We Predict Dependencies Using Domain information?	55
<i>Amir Aryani, Fabrizio Perin, Mircea Lungu, Abdun Naser Mahmood, and Oscar Nierstrasz</i>	

Session III – Comprehension II

Does the Documentation of Design Pattern Instances Impact on Source Code Comprehension? Results from Two Controlled Experiments	67
<i>Carmine Gravino, Michele Risi, Giuseppe Scanniello, and Genoveffa Tortora</i>	
Towards the Extraction of Domain Concepts from the Identifiers	77
<i>Surafel Lemma Abebe and Paolo Tonella</i>	
Reverse Engineering Co-maintenance Relationships Using Conceptual Analysis of Source Code	87
<i>Scott Grant, James R. Cordy, and David B. Skillicorn</i>	
Concern Localization using Information Retrieval: An Empirical Study on Linux Kernel	92
<i>Shaowei Wang, David Lo, Zhenchang Xing, and Lingxiao Jiang</i>	

Session IV – Code Search

Locating the Meaning of Terms in Source Code Research on “Term Introduction”	99
<i>Jan Nonnen, Daniel Speicher, and Paul Imhoff</i>	
Approximate Code Search in Program Histories	109
<i>Saman Bazrafshan, Rainer Koschke, and Nils Göde</i>	
Code Search via Topic-Enriched Dependence Graph Matching	119
<i>Shaowei Wang, David Lo, and Lingxiao Jiang</i>	
Measuring the Accuracy of Information Retrieval Based Bug Localization Techniques	124
<i>Matthew Beard, Nicholas Kraft, Letha Etzkorn, and Stacy Lukins</i>	

Session V – APIs

Exploring the Intent behind API Evolution: A Case Study	131
<i>Daqing Hou and Xiaojia Yao</i>	
A Framework Profile of .NET	141
<i>Ralf Lämmel, Rufus Linke, Ekaterina Pek, and Andrei Varanovich</i>	
Useful, But Usable? Factors Affecting the Usability of APIs	151
<i>Minhaz F. Zibran, Farjana Z. Eishita, and Chanchal K. Roy</i>	

Session VI - Security

Automatic Extraction of Secrets from Malware	159
<i>Ziming Zhao, Gail-Joon Ahn, and Hongxin Hu</i>	
Reverse Engineering of Protocols from Network Traces	169
<i>João Antunes, Nuno Neves, and Paulo Verissimo</i>	

An Android Security Case Study with Bauhaus	179
<i>Bernhard J. Berger, Michaela Bunke, and Karsten Sohr</i>	
An Exploratory Study of Software Reverse Engineering in a Security Context	184
<i>Christoph Treude, Fernando Figueira Filho, Margaret-Anne Storey, and Martin Salois</i>	
Session VII – Bugs	
How Long Does a Bug Survive? An Empirical Study	191
<i>Gerardo Canfora, Michele Ceccarelli, Luigi Cerulo, and Massimiliano Di Penta</i>	
A Preliminary Evaluation of Text-based and Dependency-based Techniques for Determining the Origins of Bugs	201
<i>Steven Davies, Marc Roper, and Murray Wood</i>	
Got Issues? Do New Features and Code Improvements Affect Defects?	211
<i>Daryl Posnett, Abram Hindle, and Prem Devanbu</i>	
Session VIII – Quality	
Impact of Installation Counts on Perceived Quality: A Case Study on Debian	219
<i>Israel Herraiz, Emad Shihab, Thanh H.D. Nguyen, and Ahmed E. Hassan</i>	
Monitoring Software Quality Evolution by Analyzing Deviation Trends of Modularity Views	229
<i>Tianmei Zhu, Yijian Wu, Xin Peng, Zhenchang Xing, and Wenyun Zhao</i>	
An Empirical Validation of the Benefits of Adhering to the Law of Demeter	239
<i>Yi Guo, Michael Wüersch, Emanuel Giger, and Harald C. Gall</i>	
Assessing Software Quality by Program Clustering and Defect Prediction	244
<i>Xi Tan, Xin Peng, Sen Pan, and Wenyun Zhao</i>	
Session IX – Dynamic Analysis	
Object-Based Dynamic Protocol Recovery for Multi-Threading Programs	251
<i>Bernhard Scholz and Rainer Koschke</i>	
An Entropy Evaluation Approach for Triaging Field Crashes: A Case Study of Mozilla Firefox	261
<i>Foutse Khomh, Brian Chan, Ying Zou, and Ahmed E. Hassan</i>	
Meta-Level Runtime Feature Awareness for Java	271
<i>Andrzej Olszak, Martin Rytter, and Bo Nørregaard Jørgensen</i>	
Using Dynamic Analysis and Clustering for Implementing Services by Reusing Legacy Code	275
<i>Andreas Fuhr, Tassilo Horn, and Volker Riediger</i>	

Session X – Reverse Engineering

Reverse Engineering of Mobile Application Lifecycles	283
<i>Dominik Franke, Corinna Elsemann, Stefan Kowalewski, and Carsten Weise</i>	
Reverse Engineering of Event Handlers of RAD-Based Applications	293
<i>Óscar Sánchez Ramón, Jesús Sánchez Cuadrado, and Jesús García Molina</i>	
Analyzing the Source Code of Multiple Software Variants for Reuse Potential	303
<i>Slawomir Duszynski, Jens Knodel, and Martin Becker</i>	
Reverse Engineering Feature Models from Programs’ Feature Sets	308
<i>Evelyn Nicole Haslinger, Roberto E. Lopez-Herrejon, and Alexander Egyed</i>	

Session XI – Evolution

Make it or Break it: Mining Anomalies from Linux Kbuild	315
<i>Sarah Nadi and Ric Holt</i>	
An Exploratory Study of Macro Co-changes	325
<i>Fehmi Jaafar, Yann-Gaël Guéhéneuc, Sylvie Hamel, and Giuliano Antoniol</i>	
An Exploratory Study of the Evolution of Communicated Information about the Execution of Large Software Systems	335
<i>Weiyi Shang, Zhen Ming Jiang, Bram Adams, Ahmed E. Hassan, Michael W. Godfrey, Mohamed Nasser, and Parminder Flora</i>	

Session XII – Binary Analysis and Refactoring

SmartDec: Approaching C++ Decompilation	347
<i>Alexander Fokin, Egor Derevenec, Alexander Chernov, and Katerina Troshina</i>	
Precise Static Analysis of Binaries by Extracting Relational Information	357
<i>Alexander Sepp, Bogdan Mihaila, and Axel Simon</i>	
Refactoring Traditional Forms into Ajax-enabled Forms	367
<i>Ming Ying and James Miller</i>	
An Empirical Study of Refactoring in the Context of FanIn and FanOut Coupling	372
<i>Alessandro Murgia, Roberto Tonelli, Steve Counsell, Giulio Concas, and Michele Marchesi</i>	

Session XIII – Developers and Modularization

Recommending People in Developers’ Collaboration Network	379
<i>Didi Surian, Nian Liu, David Lo, Hanghang Tong, Ee-Peng Lim, and Christos Faloutsos</i>	
Reasoning over the Evolution of Source Code Using Quantified Regular Path Expressions	389
<i>Andy Kellens, Coen De Roover, Carlos Noguera, Reinout Stevens, and Viviane Jonckers</i>	
Modularization Metrics: Assessing Package Organization in Legacy Large Object-Oriented Software	394
<i>Hani Abdeen, Stéphane Ducasse, and Houari Sahraoui</i>	

Industrial Experience Reports

Application Architecture Discovery - Towards Domain-driven, Easily-Extensible Code Structure	401
<i>Hitesh Sajnani, Ravindra Naik, and Cristina Lopes</i>	
Renovation by Machine-Assisted Program Transformation in Production Reporting and Integration	406
<i>Sava Mintchev</i>	
Reconstructing Traceability between Bugs and Test Cases: An Experimental Study	411
<i>Nilam Kaushik, Ladan Tahvildari, and Mark Moore</i>	

Doctoral Symposium

Recovering Test-to-Code Traceability Via Slicing and Conceptual Coupling	417
<i>Abdallah Qusef</i>	
In-Memory Database Support for Source Code Search and Analytics	421
<i>Oleksandr Panchenko</i>	
Extending Bug Localization Using Information Retrieval and Code Clone Location Techniques	425
<i>Matthew Beard</i>	

Tool Demonstrations

Stealthy Profiling and Debugging of Malware Trampolining from User to Kernel Space	431
<i>Jason Raber</i>	
Function Insight: Highlighting Suspicious Sections in Binary Run Traces	433
<i>Michelle Cheatham and Jason Raber</i>	
Understanding Legacy Features with Featureous	435
<i>Andrzej Olszak and Bo Nørregaard Jørgensen</i>	
iProblems - An Integrated Instrument for Reporting Design Flaws, Vulnerabilities and Defects	437
<i>Mihai Codoban, Cristina Marinescu, and Radu Marinescu</i>	
Focusing on Execution Traces Using Diver	439
<i>Del Myers and Margaret-Anne Storey</i>	
jPET: An Automatic Test-Case Generator for Java	441
<i>Elvira Albert, Israel Cabañas, Antonio Flores-Montoya, Miguel Gómez-Zamalloa, and Sergio Gutiérrez</i>	
SCOTCH: Slicing and Coupling Based Test to Code Trace Hunter	443
<i>Abdallah Qusef, Gabriele Bavota, Rocco Oliveto, Andrea De Lucia, and David Binkley</i>	
Author Index	445