










Keynote

- 1  **Intensions Are a Key to Program Comprehension**
(*Václav Rajlich*)
-

Impact Analysis and Changes





- 10  **Variable Granularity for Improving Precision of Impact Analysis**
(*Maksym Petrenko, Václav Rajlich*)
- 20  **Automatically Identifying Changes That Impact Code-to-Design Traceability**
(*Maen Hammad, Michael L. Collard, Jonathan I. Maletic*)
- 30  **Automatic Classification of Large Changes into Maintenance Categories**
(*Abram Hindle, Daniel M. German, Michael W. Godfrey, Richard C. Holt*)
-

Static Analysis




- 40  **A Plethora of Paths**
(*Eric Larson*)
- 50  **Practical Static Analysis for Inference of Security-Related Program Properties**
(*Yin Liu, Ana Milanova*)
- 60  **Impact Analysis and Visualization Toolkit for Static Crosscutting in AspectJ**
(*Dehua Zhang, Ekwa Duala-Ekoko, Laurie Hendren*)



Cognitive and Other Issues




- 70  **BugFix: A Learning-Based Tool to Assist Developers in Fixing Bugs**
(*Dennis Jeffrey, Min Feng, Neelam Gupta, Rajiv Gupta*)
- 80  **Resumption Strategies for Interrupted Programming Tasks**
(*Chris Parnin, Spencer Rugaber*)
- 90  **Using Activity Traces to Characterize Programming Behaviour Beyond the Lab** (*short paper*)
(*Gail C. Murphy, Petcharat Viriyakattiyaporn, David Shepherd*)
- 95  **An *in-vivo* Study of the Cognitive Levels Employed by Programmers During Software Maintenance** (*short paper*)
(*Tara Kelly, Jim Buckley*)
-

Visualization and Sonification




- 100  **Trace Visualization for Program Comprehension: A Controlled Experiment**
(*Bas Cornelissen, Andy Zaidman, Arie van Deursen, Bart van Rompaey*)
- 110  **Using Spoken Text to Aid Debugging: An Empirical Study**
(*Andreas Stefik, Ed Gellenbeck*)
- 120  **Sonification Design Guidelines to Enhance Program Comprehension**
(*Khaled Hussein, Eli Tilevich, Ivica Ico Bukvic, SooBeen Kim*)



Features and Concerns




- 130  **Instrumenting Time-Sensitive Software for Feature Location**
(*Dennis Edwards, Norman Wilde, Sharon Simmons, Eric Golden*)
- 138  **Crosscutting Patterns and Design Stability: An Exploratory Analysis**
(*Eduardo Figueiredo, Bruno Silva, Claudio Sant'Anna, Alessandro Garcia, Jon Whittle, Daltro Nunes*)
- 148  **On the Role of the Nouns in IR-Based Traceability Recovery**
(*Giovanni Capobianco, Andrea De Lucia, Rocco Oliveto, Annibale Panichella, Sebastiano Panichella*)
-

Source Code Reading







- 158  **To CamelCase or Under_score**
(*Dave Binkley, Marcia Davis, Dawn Lawrie, Christopher Morrell*)
- 168  **Reading the Documentation of Invoked API Functions in Program Comprehension**
(*Uri Dekel, James D. Herbsleb*)
- 178  **The Effectiveness of Source Code Obfuscation: An Experimental Assessment**
(*Mariano Ceccato, Massimiliano Di Penta, Jasvir Nagra, Paolo Falcarin, Filippo Ricca, Marco Torchiano, Paolo Tonella*)



Architecture and Design







- 188  **The Loss of Architectural Knowledge During System Evolution: An Industrial Case Study**
(*Martin Feilkas, Daniel Ratiu, Elmar Jürgens*)
- 198  **D_n -Based Architecture Assessment of Java Open Source Software Systems**
(*Alexander Serebrenik, Serguei Roubtsov, Mark van den Brand*)
- 208  **Standing on the Shoulders of Giants — A Data Fusion Approach to Design Pattern Detection**
(*Günter Kniesel, Alexander Binun*)
-

Short Papers I



- 218  **An Exploratory Study on Assessing Feature Location Techniques**
(*Meghan Revelle, Denys Poshyvanyk*)
- 223  **Natural Language Parsing for Fact Extraction from Source Code**
(*Jens Nilsson, Welf Löwe, Johan Hall, Joakim Nivre*)
- 228  **An Empirical Exploration of Regularities in Open-Source Software Lexicons**
(*Derrin Pierret, Denys Poshyvanyk*)
- 233  **Vector Space Analysis of Software Clones**
(*Scott Grant, James R. Cordy*)
- 238  **CnP: Towards an Environment for the Proactive Management of Copy-and-Paste Programming**
(*Daqing Hou, Patricia Jablonski, Ferosh Jacob*)
- 243  **Syntax Tree Fingerprinting for Source Code Similarity Detection**
(*Michel Chilowicz, Etienne Duris, Gilles Roussel*)



Short Papers II







- 248  **Methods for Selecting and Improving Software Clustering Algorithms**
([Mark Shtern](#), [Vassilios Tzerpos](#))
- 253  **Supporting Task-Oriented Navigation in IDEs with Configurable HeatMaps**
([David Röthlisberger](#), [Oscar Nierstrasz](#), [Stéphane Ducasse](#), [Damien Pollet](#), [Romain Robbes](#))
- 258  **A Case for Concept Programs**
([Reinhard Schauer](#), [Rudolf K. Keller](#))
- 263  **Profile-Based Type Reconstruction for Decompilation**
([K. Troshina](#), [A. Chernov](#), [A. Fokin](#))
- 268  **An Empirical Study on the Comprehension of Stereotyped UML Class Diagram Layouts**
([Bonita Sharif](#), [Jonathan I. Maletic](#))
- 273  **Who Can Help Me with This Change Request?**
([Huzefa Kagdi](#), [Denys Poshyvanyk](#))
-

Working Sessions





- 278  **Using Eye-Tracking to Understand Program Comprehension**
([Yann-Gaël Guéhéneuc](#), [Huzefa Kagdi](#), [Jonathan I. Maletic](#))
- 280  **TDD = Too Dumb Developers? Implications of Test-Driven Development on Maintainability and Comprehension of Software**
([Marco Torchiano](#), [Alberto Sillitti](#))



Tool Demonstrations

- 283  **OGAN: Visualizing Object Interaction Scenarios Based on Dynamic Interaction Context**
(*Satoshi Munakata, Takashi Ishio, Katsuro Inoue*)
- 285  **CRISTA: A Tool to Support Code Comprehension Based on Visualization and Reading Technique**
(*Daniel Porto, Manoel Mendonça, Sandra Fabbri*)
- 287  **Kenyon-Web: Reconfigurable Web-Based Feature Extractor**
(*Sunghun Kim, Shivkumar Shivaji, E. James Whitehead Jr.*)
- 289  **Prototyping Synchronization Policies for Existing Programs**
(*Y. Huang, L.K. Dillon, R.E.K. Stirewalt*)
- 291  **Proposing a Visual Approach to Support the Characterization of Software Comprehension Activities**
(*Glauco de F. Carneiro, Manoel Mendonça, Rodrigo Magnavita*)
- 293  **SODBeans**
(*Andreas Stefik, Andrew Haywood, Shahzada Mansoor, Brock Dunda, Daniel Garcia*)
-

Posters

- 295  **Design Pattern Directed Clustering for Understanding Open Source Code**
(*Zhixiong Han, Linzhang Wang, Liqian Yu, Xin Chen, Jianhua Zhao, Xuandong Li*)
- 297  **A Bug You Like: A Framework for Automated Assignment of Bugs**
(*Olga Baysal, Michael W. Godfrey, Robin Cohen*)
- 299  **Creating Task-Based Concern Maps by Merging Concern Fragments**
(*Sukanya Ratanotayanon, Susan Elliott Sim*)
- 301  **Towards Pie Tree Visualization of Graphs and Large Software Architectures**
(*Mireille Samia, Michael Leuschel*)

*Posters continued ...*

- 303  **Structure Transition Graphs: An ECG for Program Comprehension?**
(*Susan Elliott Sim, Sukanya Ratanotayanon, Leyna Cotran*)
- 305  **Enabling More Precise Dependency Analysis in Event-Based Systems**
(*Daniel Popescu, Joshua Garcia, Nenad Medvidovic*)
- 307  **Observation of Open Source Programmers' Information Seeking**
(*Khaironi Yatim Sharif, Jim Buckley*)
- 309  **Creating and Maintaining Tutorials with DEFT**
(*Andreas Bartho*)
- 311  **Improving Program Comprehension by Enhancing Program Constructs: An Analysis of the Umple Language**
(*Andrew Forward, Timothy C. Lethbridge, Dusan Brestovansky*)
- 313  **Capturing Java Naming Conventions with First-Order Markov Models**
(*Erik Linstead, Lindsey Hughes, Cristina Lopes, Pierre Baldi*)
- 315  **Automatic Detection of Internal Queues and Stages in Message Processing Systems**
(*Suman Karumuri, Steve Reiss*)
- 317  **TaskBoard: Tracking Pertinent Task Artifacts and Plans**
(*Chris Parnin, Carsten Görg, Spencer Rugaber*)
- 319  **Representing Source Code with Granular Hierarchical Structures**
(*Benjapol Auprasert, Yachai Limpiyakorn*)