



WCRE is the premier research conference on the theory and practice of recovering information from existing software and systems. WCRE explores innovative methods of extracting the many kinds of information that can be recovered from software, software engineering documents, and systems artifacts, and examines innovative ways of using this information in system renovation and program understanding.

### **GENERAL CHAIRS**

**Jim Buckley** University of Limerick, Ireland

Elliot Chikofsky Engineering Management & Integration, USA

### **PROGRAM CHAIRS**

Martin Pinzger Delft University of Technology, The Netherlands Denys Poshyvanyk The College of William and Mary, USA

#### **DOCTORAL SYMPOSIUM**

Michele Lanza University of Lugano, Switzerland

**Radu Marinescu** University of Timisoara, Romania

## **WORKSHOP CHAIRS**

Michael Collard The University of Akron, USA

Jens Knodel Fraunhofer IESE, Germany

## **TOOL DEMO CHAIRS**

Mircea Lungu University of Berne, Switzerland

Huzefa Kagdi Winston-Salem State University, USA

## **PUBLICITY CHAIR**

Michaela Greiler Delft University of Technology, The Netherlands





http://www.cs.wm.edu/semeru/wcre2011

October 17th - 20th, 2011





**KEYNOTES** 

#### Professor Mike Hinchey

Evolving Critical Systems

#### Mr Philip H. Newcomb

Architecture-Driven Modernization of the European Air Traffic Management System

### FULL TECHNICAL PAPERS (TITLES)

- Make it or Break it: Mining Anomalies in Linux Kbuild
- Object-Based Dynamic Protocol Recovery for Multi-Threading Programs
- Recommending People in Developers' Collaboration Network
- Automatic Extraction of Secrets from Malware
- Automatic Segmentation of Method Code into Meaningful Blocks to Improve Readability
- ▶ Locating the Meaning of Terms in Source Code
- Requirements Traceability for Object Oriented Systems by Partitioning Source Code
- > Approximate Code Search in Program Histories
- ▶ A Novel Analysis of Co-change
- ▶ Incremental Code Clone Detection: A PDG-based Approach
- SmartDec: approaching C++ decompilation
- Can we predict dependencies using domain information?
- Reverse Engineering of Mobile Application Lifecycles
  Assessing the Doc. of Design Patterns on Code
- Comprehension: Two Controlled Experiments
- ► How Long does a Bug Survive? An Empirical Study
- Reverse Engineering of Event Handlers of RAD-Based Applications
- A preliminary evaluation of text-based/dependency-based techniques for determining the origin of bugs
- Impact of Installation Counts on Perceived Quality: A Case Study on Debian
- Monitoring Software Quality Evolution by Analyzing Deviation Trends of Modularity Views
- ➤ An Expl. Study of the Evolution of Communicated Info. about the Execution of Large Systems
- Towards the Extraction of Domain Concepts from the Identifiers
- Precise Static Analysis of Binaries by Extracting Relational Information
- ▶ Reverse Engineering of Protocols from Network Traces
- An Entropy Evaluation Approach for Triaging Field Crashes: A Case Study of Mozilla Firefox
- Exploring the Intent behind API Evolution: A Case Study On the Effectiveness of Simhashing in Clone Detection on Large Scale Software System
- An empirical study of framework design and usage for .NET

# SHORT TECHNICAL PAPERS (TITLES)

- An Empirical Validation of the Benefits of Adhering to the Law of Demeter
- ➤ An Investigation into the Impact of Software Licenses on Copyand-Paste Reuse in OSS Projects
- Reasoning over the evolution of source code with quantified regular path expressions
- ▶ Refactoring Traditional Forms into Ajax-enabled Forms
- Meta-Level Runtime Feature Awareness for Java
- Analyzing the Source Code of Multiple Software Variants for Reuse Potential
- Using Dynamic Analysis and Clustering for Implementing Services by Reusing Legacy Code
- ▶ An Android Security Case Study with Bauhaus
- An Empirical Study of Refactoring in the Context of FanIn and FanOut Coupling
- Reverse Engineering Feature Models From Programs' Feature Sets
- Assessing Software Quality by Program Clustering and Defect Prediction
- Modularization Metrics: Assessing Package Org. in Legacy Large Object-Oriented Software
- Code Search via Topic-Enriched Dependency Graph Matching
- Reverse Engineering Co-maintenance Relationships Using Conceptual Analysis of Code
- Concern Localization using Information Retrieval: An Empirical Study on Linux Kernel
- Measuring the Accuracy of Information Retrieval Based Bug Localization Techniques
- ▶ Got Issues? Do New Features and Code Improvements Affect Defects?
- Internet-scale Real-time Code Clone Search via Multi-level Indexing
- Useful, but usable? Factors Affecting the Usability of APIs
- An Exploratory Study of Software Reverse Engineering in a Security Context

