# SAC'03 18th Annual ACM Symposium on Applied Computing

## Special Track on Software Engineering: Applications, Practices and Tools

Stefan Gruner, University of Southampton, Department of ECS
http://www.dsse.ecs.soton.ac.uk/

**Introduction:** The *Software Engineering: Applications, Practices and Tools* track at the *18th Annual ACM Symposium on Applied Computing* (March 9-12, Melbourne, Florida, USA) was the melding of original track proposals on Software Testing, Formal Methods and Software Interoperability. The result was a very wide appeal for papers which spanned these topics and several others. After the merging of these three smaller tracks, the goal of our track then became to solicit high quality submissions that span the broader field of software engineering. As this is an applied conference, our appeal focused on papers that had a practical flavor and demonstrated results of the approach they presented. We were very pleased in general with the quality of the submitted papers. In total, the call for papers generated 38 submissions. Papers were received from 15 different countries including the US, Spain, France, Belgium, China, Finland, Brazil, Great Britain, and several others. There were 10 members of the original program committee, who graciously agreed to serve and review submissions. The abundance of submissions required us to solicit additional reviewers to cope with the overflow. In the end, there were 118 reviews generated by 26 reviewers. Of the 38 papers submitted, 12 were strongly recommended for inclusion in the track. There were several papers though that were also rated highly by some reviewers but were not able to be included due to space limitations. Below is a brief description of the accepted papers, which can all be found in the SAC'03 Proceedings published by the ACM.

**Formal Semantics and Design Pattern Issues in UML and Java:** This group of papers addresses important current issues in the application of formal methods and design patterns in the system and software design. Two different approaches of providing a formal semantics for UML state charts are included. Another UML paper proposes a framework for the preservation of design patters in applied UML designs and their composition. The forth paper discusses issues in the Composition of Java Frameworks Controlflows.

- *An Integrated Framework for Formal Development of Open Distributed Systems* by Issa Traore, Demissie B. Aredo and Hong Ye. Formal System development still lacks a standardized, widely accepted formalism. The authors of this paper propose an integrated framework based on the Unified Modeling Language (UML) and the Prototype Verification System (PVS). A formal semantics of UML state charts in PVS notation is presented. A case study - the specification of a complete high-speed serial protocol using the CASE tool PrUDE - demonstrates this approach.

- *Modeling the Meaning of Transitions from and to Concurrent States in UML State Machines* by Egon Boerger, Alessandra Cavarra and Elvinia Riccobene. This second approach of formalizing the semantics of UML state charts is based on Abstract State Machine (ASM) models. In this paper, the authors complete their in-depth work in this area. A key feature of this method is modularity, which allows the insertion of new submachines in already existing UML state diagrams.

- *Representing the Applications and Compositions of Design Patterns in UML* by Jing Dong. Another important open problem in

UML application is addressed in this contribution: How can UML design patterns can still be recognized, if they are applied in a concrete situation or if several such patterns are composed? The authors present a framework of notations that keeps track of the pattern information and make patterns identifiable and traceable.

- *On the Composition of Java Frameworks Control-Flows* by Ana C. V. de Melo and Bruno M. Moutinho. This paper presents a study on the composition of Java frameworks control-flows, where a third framework is created from two existing ones. The study illustrates potential problems that may appear as two control-flows are composed through message passing.

**Software Testing:** The group of papers below spans many aspects of software testing. The first two are devoted to automatic test generation. The next two seek measures of testing effectiveness in the general sense. The fifth paper presents a testing methodology bases on the search for the violation of a specific safety-critical property. The final paper explores the application of Test Driven Design through a case study.

- *A method for the automatic generation of test suites from object models* by Jim Davies, Charles Crichton and Alessandra Cavarra. This paper explains how object-oriented specifications, written in the Unified Modeling Language (UML) can be translated into formal, behavioral descriptions and used as a basis for automatic test generation.

- *Towards Model-Based Generation of Self-Priming and Self-Checking Conformance Tests for Interactive Systems* by Amit Paradkar. This paper presents a method for generating conformance tests for interactive systems by extending the Category-Partitioning testing method and the associated Test Specification Language.

- *Using Operational Distributions to Judge Testing Progress* by Elaine J. Weyuker. This paper presents the criteria of field usage coverage to judge software testing progress. The underlying motivation is that other testing criteria may not be particularly informative when assessing the number of defects encountered in the field after the software is released.

- *On the Use of the Classification-Tree Method by Beginning Software Testers* by Y. T. Yu, S. P. Ng, Pak-Lok Poon and T. Y. Chen. This paper describes a case study through which the authors attempt to understand the methods used by novice software testers to test their own programs, as well as their perception of the classification-tree method. The authors present a convincing trial implementation of the method and the participants benefited by the implementation of the method to their own application testing.

- *Property-Oriented Testing: A Strategy for Exploring Dangerous Scenarios* by Olfa Abdellatif-Kaddour, Pascale Thevenod-Fosse and Helene Waeselynck. This paper introduces a method for testing for breach of a specific property. The method is appropriate for identifying violations of critical, safety related properties of the software and the case study presented shows its application to control software of a boiler with the property of interest being non-explosion.

- *An Initial Investigation of Test Driven Development in Industry* by Boby George and Laurie Williams. This paper focuses on the application of Test Driven Development (TDD) and its impact on the quality of software produced. The authors present a compelling differential cases study between two groups, one using TDD and the other using the traditional waterfall method.

**The Wider Software Engineering Discipline:**

- *A Methodology to Provide and Use Interchangeable Services* by Brian Fenicle and Tim Wahls As the title suggests, this paper introduces a framework for pay per use software that encompasses issues such as vendors which offer similar services and an approach to integrate these to allow the user to make decisions about providers but still make the implementation nuances that follow transparent to the user.

- *Efficient Mediators through Dynamic Code Generation: a Method and an Experiment* by Yasushi Shinjo, Toshiyuki Kubo and Calton Pu. This paper presents an implementation method of mediators that dynamically generate dynamic interfaces at runtime. The implementation method is based on the concept of closures. The authors evaluate the strengths of their method by comparing it with other existing methods in terms of code size, expressiveness and applicability, and performance.

**Conclusion:** We, the track chairs, are convinced that these 12 papers represent one of the strongest offerings of the Software Engineering track to date and represent a worthwhile contribution to the field. Because of the good success of our track at SAC'03, we are planning to continue the story at SAC'04 next year — watch out for our Call for Papers!

- Track Chair: *Herbert H. Thompson*, Department of Computer Sciences, Florida Institute of Technology, 150 W. University Blvd. Melbourne, Florida 32901 USA. E-Mail: hthompson@se.fit.edu. Phone: ++1 321 795 4531.
- Co-Chair: *Chia-Chu Chiang*, Department of Computer Science, University of Arkansas at Little Rock, 2801 S. University Ave. Little Rock, Arkansas 72204-1099 USA. E-Mail: cxchiang@ualr.edu. Phone: ++ 1 501 569 8142. Fax: ++ 1 501 569 8144.
- Co-Chair: *Ibrahim K. El-Far*, Computer Sciences Department, Florida Institute of Technology, 150 W. University Blvd. Melbourne, Florida 32901 USA. E-Mail: ielfar@acm.org. Phone: ++ 1 321 223 8978. Fax: ++ 1 321 674 7046.
- Co-Chair: *Stefan Gruner*, Dept. of Electr. and Comp. Science, University of Southampton, Southampton, Hampshire SO17 1BJ Great Britain. E-Mail: sg@ecs.soton.ac.uk. Phone: ++ 44 23 80 59 31 23. Fax: ++ 44 23 80 59 30 45.
- Co-Chair: *Markus Montigel*, Computer Science Department, University of New Orleans, New Orleans, Louisiana 70148 USA. E-Mail: markus@cs.uno.edu. Phone: ++ 1 504 280 7075. Fax: ++ 1 504 280 7228
- Co-Chair: *James A. Whittaker*, Computer Sciences Department, Florida Institute of Technology, 150 W. University Blvd. Melbourne, Florida 32901 USA. E-Mail: jw@se.fit.edu. Phone: ++ 1 321 674 7638. Fax: ++ 1 321 674 7046.