Advances in Software Specification and Verification
Introduction to Minitrack

Ann E. Kelley Sobel
Computer Science & Systems Analysis Department
Miami University
Oxford, OH 45056
sobelae@muohio.edu

Modern society increasingly depends on complex software systems for operating its critical infrastructures, including transportation, communication, finance, energy distribution, and healthcare. As dependency grows, the consequences of system failures become increasingly severe. Growing recognition of risks and the consequences of failures is driving increasing use of rigorous specification and verification techniques for such systems.

The Advances in Software Specification and Verification minitrack focuses on research and applications that will drive widespread use of rigorous specification and verification technologies, particularly for complex, large-scale systems. The minitrack encourages papers on new specification and verification techniques, methods for scale-up to large systems, techniques for complexity reduction in specification and verification, integration of specification and verification methods, development of engineering practices and tools, and case studies.

The three papers accepted for this year’s minitrack focus on promising research areas in system specification and verification. The papers cover a number of emerging research topics, including incorporation of XML-based user documentation in testing to both validate the documentation and test the software, a novel application of control theory methods to prove key self-stabilizing properties of distributed systems, and use of linear constraints to exploit parametric real-time model checking capabilities. At least three external reviewers refereed each of the submitted papers. The accepted papers are summarized below.

In today’s rapid, Internet-time development environment, user documentation is often seen as a low priority task that takes resources away from primary development activities. However, if documentation were written in a machine-interpretable form, the software behavior it describes could be automatically verified in testing. Documentation tasks would not represent an added burden, but rather an integral part of the development process, and errors in documentation would be treated on a par with errors in software. In the paper Software Verification and Functional Testing with XML Documentation, author Ernest J. Friedman-Hill develops and illustrates a method for achieving this objective. XML-based user documentation becomes interpretable as part of a system’s specification, and evaluation of software conformance to the specification simultaneously verifies the documentation and serves as a functional test of the software. The paper presents an application of these ideas to user documentation for the Java Expert System Shell, a programming library that supports a scripting language.

A distributed computer application can often be made more robust by incorporating a self-stabilization property that guarantees it will eventually return to and remain within a specified set of states, regardless of the starting state. Such self-stabilizing distributed systems can potentially recover from transient failures. Unfortunately, proof of self-stabilization properties can be extremely difficult with current techniques. In the paper A New Verification Technique for Self-Stabilizing Distributed Algorithms Based on Variable Structure Systems and Ljapunov Theory, author Oliver Theel describes how methods from control theory can be applied to simplify self-stabilization proofs. In particular, application of century-old techniques from Ljapunov theory yields a new proof method. Verification of self-stabilization is reduced to identification of a Ljapunov function for which stability is guaranteed. A substantial body of methods exists for constructing such functions.

The IEEE 1934 serial bus standard enables fast transfer of high-bandwidth, multimedia data streams among plug-and-play devices including PCs, camcorders, video disks, and music systems. The protocol involves both real-time and probabilistic aspects. Parts of IEEE 1934 have been formally specified and/or verified, however, its Root Contention Protocol has not been verified using model checking techniques. In their case-study paper Application of Parametric Model Checking – The Root Contention Protocol, authors G. Bandini, R. L. Speelberg, R. C. H. de Rooij, and W. J. Toetenel describe verification of the protocol using the PMC model checker. The results show that probabilistic aspects of the protocol could be successfully dealt with in the model checking process.