

Testing and Certification of Trustworthy Systems Introduction to Minitrack

Alan R. Hevner
*Information Systems &
Decision Sciences Dept.
Univ. of South Florida
Tampa, FL 33620
ahevner@coba.usf.edu*

Richard C. Linger
*CERT Research Center
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213
rlinger@sei.cmu.edu*

Gwendolyn H. Walton
*Dept. of Mathematics &
Computer Science
Florida Southern College
Lakeland, FL 33801
ghwalton@flsouthern.edu*

The specification, development, and certification of trustworthy computing systems present substantial research challenges. Modern society is increasingly dependent on large-scale network systems for operating its critical infrastructures, including transportation, communication, energy, commerce, healthcare, and defense. As a result, the consequences of failures are becoming increasingly severe. These systems are characterized by heterogeneous distributed computing, high-speed networks, limited visibility and control, and the combinatorial complexity of asynchronous behavior. As a result, effective methods for testing and certification of trustworthy systems are in great demand. The Testing and Certification of Trustworthy Systems Minitrack provides a venue for research results and contributions to their practical application in the software systems of the future. Specific topic areas include:

- New techniques for trustworthiness certification
- Testing and certification metrics and measures
- Testing attributes such as security and survivability
- Engineering practices and tools for certification
- Testing in system maintenance and evolution
- Specification methods to support system certification
- Role of correctness verification in system certification
- Industrial case studies in testing and certification

The Minitrack papers to be presented during HICSS-38 are summarized below.

In their paper *Data Assurance in Conventional File Systems*, authors S. Rudan, A. Kovacevic, D. Jovic, V. Milutinovic, S. Selkirk, and C. Milligan discuss the problem of protecting file data from accidental or malicious modification. They apply public/private keys and hash value encryption in a SmartCard environment to help ensure data consistency and security.

In his paper *The Refined Algorithm ReCDRG to Construct DRG Graph for the Object-Oriented Class-Level Testing*, author H. Chen discusses a methodology for selecting fundamental pairs of equivalent ground terms as test cases for testing object-oriented software. Data member Relevance Graphs (DRGs) are employed to determine the observational equivalence of two objects generated by executing the test cases. A refinement to the algorithm for constructing DRGs is presented.

In *Perspectives on Redundancy: Applications to Software Certification*, authors A. Mili, F. Sheldon, F. Mili, M. Shereshevsky, and J. Desharnais observe that system redundancy, ordinarily discussed in the context of fault tolerance, can be employed to detect, diagnose, and correct errors in system operation. Three views of redundancy are presented and foundations for analyzing redundancy are explored as a basis for enhancing the design of fault tolerant systems.

In *Simulation-Based Validation Tools for Distributed Network Protocols*, authors K. Ravindran, K. Kwait, and G. Ding introduce state-machine-based validation modeling to capture external interface constraints, environmental perturbations, and the internal rules and procedures of protocols. Discrete event simulators can then be used to check protocols for safety and liveness properties in given environments. A case study of voting protocols under complex failure modes is presented.

In their paper *High Volume Software Testing Using Genetic Algorithms*, authors D. Berndt and A. Watkins explore strategies for combining automated test suite generation based on genetic algorithm methods with high-volume, long-sequence testing to make this approach more scalable. These methods have proven effective in detecting and correcting errors in component coordination and resource consumption.

Author S. Prowell, in the paper *Using Markov Chain Usage Models to Test Complex Systems*, introduces the application of concurrency operators to test cases generated from simple Markov chain usage models. Because the operators are applied to test cases, not usage models, analysis of the underlying models is preserved. This approach permits creation of sophisticated test cases for systems with multiple independent streams, while avoiding the problem of state explosion.