Ionut Buricea

Ionuţ Buricea

phone: (650) 967-6253
email: ionutb@cis.ksu.edu
URL: http://www.cis.ksu.edu/~ionutb/

Research Interests

Modular design and compositional verification of communication protocols, reachability analysis of timed automata, model-checking techniques for timed protocols.

For a long time, my interest has been focused on studying formal specification and composition of communication protocols and services. For my diploma project in Romania I implemented a high-level description of the TCP protocol using a C implementation of the ISO standard formal description technique ESTELLE for distributed systems. During the first two years of my PhD program at Kansas State University I studied techniques for composing untimed communication protocols expressed as Finite State Machines. In these techniques, protocols are represented at the more compact and higher level of service specification (ss) or at the closer to implementation level of protocol specification (ps). In published material we show a strategy in which we allow composition to be specified at the service specification level (that is, $ss(P)$ and $ss(Q)$ are first combined to obtain $ss(R)$). Given $ss(R)$, we provide an algorithm to mechanically combine $ps(P)$ and $ps(Q)$ to generate $ps(R)$ such that $ps(R)$ satisfies $ss(R)$. We show that analysis of $ss(R)$ is sufficient to ensure that $ps(R)$ satisfies certain safety and liveness properties. This results in efficient validation as state space of $ss(R)$ is typically significantly smaller than that of $ps(R)$.

Trying to extend this work to timed protocols, I studied the verification of timed systems expressed as Timed Extended Finite State Machines. We developed a reachability analysis algorithm for parallel composition of such machines.

Education

PhD (computer science), August 1996 - December 2003, Kansas State University. GPA 4.0.

MS (computer science), 1996, “Politehnica” University of Bucharest, Romania. GPA 4.0.

BS (computer science), 1995, “Politehnica” University of Bucharest, Romania. GPA 3.5.

Experience

Areas of Experience

Programming Languages: C, C++, Java, ML, Pascal, Lisp, ELF meta-language, Unix shell script(csh, bash, tcsh)

Specialised Languages and Tools: HTML, XML, SQL, mSQL, ORACLE, TCP/IP, yacc/lex (bison/flex)

Specification Languages and Formalisms: Linear Temporal Logic, Promela, Estelle

Verification Tools: Spin model checker

Operating Systems: Solaris, Linux, Windows NT

Work Experience

Software Developer, Spring 2004, Intellisys, Santa Clara, California.

Research Assistant, Fall 1996 - Summer 1998, Department of Computing and Information Sciences, Kansas State University. Professor: Gurdip Singh.

Computer Engineer, January 1996 - July 1996, Institute for Computing Technologies,Bucharest, Romania. Duties included: writing Unix shell scripts to automate formatting of technical documentation files in man page style, using nroff; configuring and maintaining the sendmail.

Research and Implementation Projects

Spring 2004 Development of a light-weight profiler for Java applications, essentially by instrumenting the Java byte-code. The profiler accounts for time spent by the garbage collector as well. To this end, technologies such as JDI, JNI, JVMPI and Javassist are used.

Autumn 2002 An ML Implementation of a Reachability Algorithm for Timed Extended Finite State Machines (based on paper 1. in the Publications section). For my implementation, I used a model checker written as part of the Bandera software model checking project at Kansas State University (http://bandera.projects.cis.ksu.edu/). Considering a system consisting of a set of Timed Extended Finite State Machines which interact with one another through different kinds of synchronisation constraints specified as shared boolean variables appearing in transition guards, the implemented algorithm is designed to verify if the synchronisation constraints are consistent with the real-time specifications of the Timed Machines in the system. The algorithm works in discrete time, it avoids state generation for every tick of the clock, and it reduces path interleavings by choosing independent transitions, one in each automaton, to trigger in parallel at every step in the analysis. The implementation consists of about 2500 lines of Standard ML code. I evaluated my implementation on extensive case studies.

Spring 1999 Semester Project: A meta-engine web site which takes a query and puts together the results from various well-known search engines. It was implemented using servlets and JavaWebServer, and it also featured an ORACLE database server running in the back end.

Fall 1997 Research Project: Before I shifted my research to automated protocol verification, I investigated the design of coordination services (written as Java generic frameworks) of distributed applications, intended to implement synchronisation-related requirements for a group of communicating processes, typically involved in a collaborative interaction. Among other things, the DiCE project at Kansas State University continues and developes ideas born during that period of time (http://www.cis.ksu.edu/∼singh/coordination.html).

Spring 1997 A generic replicated workers framework written in Java using Java interfaces. Designed as a generic coordination abstraction for a broad class of naturally concurrent problems, using Java threads. Instantiated for solving numerical integration problems.

Spring 1995 Diploma Project: An Implementation of the TCP Protocol in High-Level Specification Language Estelle.

Spring 1993 Semester Project: A full-fledged, performant text editor featuring regular expression search capability, file buffering, direct manipulation of the screen memory. Written in C++. Done in a team of three.

Teaching Experience

Teaching Assistant, Spring 2002 and 2001, Department of Computing and Information Sciences, Kansas State University, for CIS 725 “Advanced Computer Networks”. Professor: Gurdip Singh.

Teaching Assistant, Fall 2001 and 2002, Department of Computing and Information Sciences, Kansas State University, for CIS 570 “Introduction to Formal Language Theory”. Professor: Allen Stoughton.

Teaching Assistant, Fall 1998 - Fall 2000, Department of Computing and Information Sciences, Kansas State University, for various courses, including CIS726 “Advanced WWW Technologies”, CIS825 “Topics in Distributed Systems”, CIS826 “Protocol Engineering”. Professors: Gurdip Singh, Daniel Andresen

Relevant Classes

Specification and Verification of Reactive Systems- Fall 1997. Linear Temporal Logic (LTL) as a specification formalism for liveness and safety properties of programs; the verification tool Spin and language Promela for detecting invariant violations, deadlocks, livelocks, etc.; as a semester project, I verified various properties of a distributed algorithm which insures synchronisation of multiple concurrent distributed processes, using Spin and Promela.

Protocol Design and Engineering - Spring 1997; Concepts of protocol design, specification languages and formal description techniques, safety and liveness properties, protocol validation, protocol synthesis, protocol translation, implementation and conformance testing.

Analysis of Algorithms - Fall 1996. Complexity analysis, P and NP complexity classes.

Advanced Programming Languages - Spring 1998. Topics included: lambda calculus, data typing, operational semantics, denotational semantics; practical projects included programming in ELF, a meta-language that enables the specification and design of programming languages.

Advanced Computer Networks - Fall 1996; among other things, it included writing a Sliding-Window Data-Link Layer protocol, involving socket programming in C.

Advanced Compiler Design and Optimisation - Spring 1998; topics included data flow analyses, program transformation for state-space reduction.

Operating Systems - 1997. Interprocess communication in C under Unix (fork, signals, shared memory); Hoare-style proofs of correctness (pre and post-conditions, proofs of non-interference), synchronisation techniques (mutexes, semaphores, critical regions, barriers, monitors).

Other Activities

Summer 1998, attended International Summer School Marktoberdorf, Germany - "Calculational System Design". Lectures included "A Generic Compositional Reachability Static Analyzer for an Imperative Language (by Abstract Interpretation)" by Patrick Cousot, "The Analysis of Security Protocols, Based on the Dolev-Yao Model" by John Mitchell, "Binary Decision Diagrams and the CUDD Tool Applied to the Verification of Sequential Circuits" by Fabio Somenzi, and "Formal Description of Telecommunication Services using Promela and Z" by Pamela Zave.

Referee for International Conference on Computer Aided Verification, 2002; Journal of TCS, 2004.

Publications

Refereed Papers

1. Buricea, I., Singh, G., *Compositional Design of Distributed Real-Time Protocols*, in **Proceedings of the 12-th IASTED International Conference on Parallel and Distributed Computing and Systems**, November, 2000.
2. Singh, G., Buricea, I., Mao, Z., *Composition of Service Specifications*, in **Proceedings of the 6-th IEEE International Conference on Network Protocols**, October 1998.

Other Papers

Buricea, I., *Analysing Composed Real-Time Protocols*, 2003 (PhD. Thesis)

Buricea, I., Haack, C.H., Pasareanu, C.S., *Using the Smash Product to Compose Data Flow Analyses*, 1998, www.cis.ksu.edu/˜ionutb/801/paper.ps

References

Dr. Matthew Dwyer
324E Nichols Hall, Manhattan, KS 66506
Dept. of Computer and Information Sciences
Kansas State University
Email: dwyer@cis.ksu.edu

Dr. Mitch Neilsen
219D Nichols Hall, Manhattan, KS 66506
Dept. of Computer and Information Sciences
Kansas State University
Email: neilsen@cis.ksu.edu

Dr. Gurdip Singh
324D Nichols Hall, Manhattan, KS 66506
Dept. of Computer and Information Sciences
Kansas State University
Email: singh@cis.ksu.edu