Research Statement

Philip A. Wilsey

I have written the following statement so that prospective students (and others) can better understand the research that I am interested. If you are seeking a graduate advisor, a senior project,[1] or simply a prospective to aid your work with another advisor, perhaps this writing will help you decide if I can help you in your quest. Please note, that while I write this document in first person singular, almost everything I do involves at least one or more student or faculty co-investigator. Finally, my door is open; stop by and spend some time.

Overview

I am an experimentalist working in distributed systems and computer aided design (CAD) of electronic systems. In distributed systems, I have been studying the application of feedback control to optimize distributed system operation. The focus of these investigations has been Parallel and Distributed Simulation (PADS) with applications to: Networks, Mixed-Technology (continuous/discrete) Systems, and Digital Systems. In CAD, I have focused my studies on improvements and analysis of the VHDL hardware modeling language. In particular, I have worked on (i) static analysis and parallel simulation of VHDL (and now VHDL-AMS), (ii) formal modeling, optimization, and mechanized reasoning over VHDL models, and (iii) object-oriented extensions for VHDL. I am beginning to work in the area of modeling and simulation of biological systems.

As an experimentalist, I believe strongly in distributing my experimental systems so that others can replicate my research results. Consequently, I have considerable experience developing and distributing large software systems. While most of my systems are directly download-able from my web pages, two packages from my research activities (SAVANT and TYVIS) are also in the Debian Linux distribution. These packages are now maintained by a commercial entity and have commercial users including Intel, Motorola, Alcatel, and Toyota.

The principle objectives of my studies are the advancement of knowledge and understanding of distributed systems and CAD by the: (i) development of pragmatic methodologies; (ii) development of prototype tool sets; and (iii) introduction into the graduate and undergraduate curricula. My intent is to better understand the world of computing and to communicate that understanding (and the process to by which one can add to that understanding) to my students and colleagues. In many cases, my understanding begins with a crude idea to improve a system; the idea is applied and analysis begins. Ideally, the experimental investigations provide new insight to the fundamental behaviors of the system so that future investigations can work from stronger fundamentals. Key to my work is careful adherence to the scientific method. Over the years, I have been surprised to learn that I need to revisit this with new students. It is something we are all taught, but something that is not always applied—heck, I can't make coffee without a hypothesis and plan of study!

Past Research Activities

A Formal Model of VLSI Systems Compatible with VHDL: This work uses an interval temporal logic to formally define a machine model that is congruent with the hardware description language VHDL. The emphasis of this project has been toward the development of a machine models that characterizes the dynamic behaviors of VHDL. The model is unique in that it is not derived from the uniprocessor simulation kernel generally used to define VHDL behaviors. The semantics of this model was embedded into the PVS proof maintenance system and we were able to reason about equivalences between two VHDL models. Thus we were able develop compiler optimizations for parallel simulation (see SAVANT below) and show their correctness with the proof maintenance system.

In addition to the dynamic model, the project also developed a formal statement of the static semantics of VHDL that shows when a VHDL model is well typed. Furthermore, a reduction algebra was developed to formally define

[1] Virtually all of my senior projects are based on some aspect of my research or teaching activities.

those structures in VHDL that are defined to be equivalent in the language standard. The results of these investigations are summarized in a book published in 1999 by Kluwer Academic.

Object-Oriented Extensions to VHDL/VHDL-AMS for Systems Level Modeling: I had the good fortune that Dr. Peter Ashenden decided to spend a his sabbatical year studying with me. This association began a series of investigations to seamlessly extend VHDL and VHDL-AMS for systems level modeling. These investigations borrowed heavily from Ada 9X (generics, inheritance) and CSP (communication channels). The chief challenge to these investigations were to develop and refine extensions that presented strong additions to the modeling capabilities of VHDL (specifically into the system level domains) and that were syntactically and semantically consistent with the existing VHDL standard (at the time VHDL-93). I have expanded these studies to VHDL-AMS with Dr. Greg Peterson and I am beginning related investigations with Dr. Bernie Zeigler (and others) into the challenges of web-based collaborate engineering of mixed-signal hardware/software systems (using a language tentatively named nVHDL).

Massively Parallel SIMD/MIMD Multiprocessing: The project investigated the validity of the assertion that multiprocessors with a small, fixed number of control units can efficiently support both the SIMD and MIMD processing paradigms. The resulting multiprocessor must efficiently support both SIMD and MIMD processing, while minimally hindering the performance in either mode. The approach will be to restrict the complexity of the processing elements (PE) and use a fixed number of control units that does not increase with the number of PEs. This problem required a careful consideration of the MIMD instruction set to be interpreted. In particular, the execute phases of the interpreted instructions must be made as reusable as possible to maintain high PE utilization. While software optimizations such as these produced speedups, we also identified a few simple, inexpensive, hardware modifications that would dramatically boost MIMD performance at virtually no cost to SIMD performance.

Current Research Interests

Optimistically Synchronized Parallel Digital System Simulation: This project addresses the use of optimistic synchronization to speed parallel digital system simulation. In particular, we are studying the use of the Time Warp Mechanism to synchronize a parallel simulator that is automatically derived from hardware designs expressed in, the hardware description language, VHDL. Preliminary versions of a VHDL compiler and parallel simulation kernel are complete. During the construction of this parallel simulator we have discovered a mechanism for efficiently realizing lazy reevaluation and have discovered a new optimization to the Time Warp mechanism called rollback relaxation. During these investigations, we have also extended Jason Lin's model of Time Warp simulations to allow a direct comparison of the execution time overheads and memory overhead of periodic checkpointing to incremental state savings.

At present, we are pursuing mechanisms for the parallel simulation of mixed-signal systems. We have been working with electronic systems modeled in VHDL-AMS and are also beginning to work with biological systems. We publihed our first results with an optimistic synchronization strategy in 1997 and are now integrating that work with a complete VHDL-AMS simulation system. In addition, we have been working with Sandia National Labs to integrate their parallel SPICE engine (Xyce) with our parallel discrete event simulator (WARPED). An additional challenge is their desire to keep the systems relatively independent. Hence we have been working with a simulation backplane similar to the RTI of the HLA. We are also in the preliminary stages work with UTK to investigate the parallel simulation of biological systems using a cluster of workstations augmented with FPGA acceleration boards. In this project, we are planning to study the dynamic migration of model related and kernel related simulation functionalities into the FPGA subsystem as performance needs dictate.

Applying Feedback Control for Online Adjustment of Programs: This project studies the application of concepts and ideas from traditional EE feedback control theory to tune running programs. In particular, we are studying the construction and use of a software sensor/control library. The idea is to deploy (parallel) software that dynamically configures itself to an ideal operating state. We are not attempting to replace good static configurations of software, but rather provide a mechanism to deploy self-tuning software when static configurations fail. We have successfully applied both non-linear and adaptive control systems into the warped parallel simulator. Currently we are investigating techniques that will allow us to make formal statements about the stability and convergence properties of controlled software.

Web-based Collaborative Design and Analysis: This ongoing project targets effective use of the various resources available via the World Wide Web (WWW) for modeling and analysis of systems. The models of the systems are composed in a hierarchical fashion using custom software modules and those offered by third party manufacturers. A simple yet robust system specification language is used to specify the models of the system. The research addresses portability, inter-operability, and proprietary information issues that arise during sharing and reuse of software modules over the WWW. A distributed simulation infrastructure was developed to enable analysis of the system models using simulation. The web-based simulation framework uses the optimistically synchronized simulation kernel to achieve distributed simulation.

The use of formal software techniques for verification and validation of the models is currently is being explored. The issues involved in enable mixed-mode simulations over the WWW is under study. Research to enable interoperation of various simulators using predefined interfaces, such as the High Level Architecture (HLA), is also underway. The project aims to tie the various distributed resources of the WWW into a giant design and analysis backbone.

Released Software

I (and my students) currently distribute and maintain several software packages (see `http://www.ececs.uc.edu`. Some of the more widely used packages are:

SAVANT and TYVIS: a freely redistributable VHDL analyzer and VHDL simulation kernel. The SAVANT analyzer is called `scram` and it translates VHDL into the IIR intermediate form of the AIRE standard (see `http://www.vhdl.org/vi/aire/index.html`. A back-end code generator for IIR outputs C++ for compilation and execution with the TYVIS kernel. Both SAVANT and TYVIS were initially developed as part of an Air Force SBIR and are now freely redistributable under the GNU LGPL license. The systems contain approximately 100,000 lines of C++ code and is used by companies worldwide, notably: Intel, Alcatel, Motorola, and Toyota as well as in numerous University research programs. Both systems have Debian, Redhat, and Solaris packages. Additional information is available at `http://www.ececs.uc.edu/~paw/savant` and `http://www.ececs.uc.edu/~paw/tyvis`.

WARPED: a general purpose, Time Warp synchronized, distributed simulation kernel. The software was first released in 1995 and has been used by industry and universities. The system contains approximately 8,000 lines of C++ code and uses any of the MPI, TCP, or MPI-BIP communication subsystems. The system is robust and our studies have shown capacities of 10 million parallel simulation objects and executions of billions of events using a small 8-node Pentium II cluster. The principle target architecture is distributed memory, but it has been ported to many platforms including shared-memory and clusters of SMPs. The system conforms to a discrete event simulation API and is currently being integrated with a continuous simulation kernel to support mixed-technology simulation. See `http://www.ececs.uc.edu/~paw/warped` for additional information.

FWNS: A Framework for Web-based Network Simulation]: The framework provides necessary infrasturcture for modeling and simulating communication networks. FWNS has been developed using a combination of C++ classes and Hyper Text Markup Language (HTML) scripts. The primary component of the framework is a FWNS server that performs the core functionalities. Access to the server is provided through Common Gateway Interface (CGI) script based HTML web pages. A Topology Specification Language (TSL) was developed to ease specification of the network topology along with the information necessary for simulation. The simulation modules for the various components constituting the network are obtained from the FWNS factories as directed by the user. The framework includes the C++ classes required to build the distributed factories. An online version of the framework along with further details on its design and usage are available at `http://www.eng.uc.edu/~dmadhava/`.

Epilogue

In general, I like to work with other investigators, be they students, faculty, or other researchers. I find that these associations expose my ideas to more critical analysis and that I end up conducting much better work as a result. To all my past collaborators, thank you; my life has been much richer because of you.