Teaching a Unified Approach to Modeling and Simulation

James C. Browne

Department of Computer Sciences, University of Texas at Austin, USA
browne@cs.utexas.edu

Presentation Summary

Modeling plays a fundamental role in the design and analysis of complex systems in general and computer systems in particular. A model is an abstraction of a system. There are frequently multiple models of a system with each model (performance model, reliability model, functional model, etc.) being specialized for analysis of a specific property type. Models, like systems, are represented in languages with each language being model/property type specific. Algorithms and methods for analysis of models are model type and representation specific. There are separate research communities and commonly separate classes for each model/property type: performance modeling, model checking for functional correctness, etc. There is a common underlying conceptual foundation for these model/property types. Each model/property type may have analytically evaluatable and executable representations. The analytical evaluatable and executable representations are commonly founded on specialized versions of state transition systems. Evaluation of models used for performance, reliability and performability evaluations are instances of Markov state transition systems. Model checking is commonly based on Kripke structures which are state transition graphs. Executable representations are evaluated with some variant of discrete event simulation.

Convergence of the models and evaluation algorithms is occurring in the research community. Performability is a unification of reliability, availability and performance. Stochastic process algebras which merge performance modeling with process algebras used for functional models of concurrent systems are emerging. Metric temporal logics integrate quantitative time into model checking. There are Petri net based modeling systems which support both functional and performance modeling. WCET is a unification of static control flow graphs (which are equivalent to state machines) and performance modeling.

The goal for the course is to combine a unified perspective on systems modeling with operation effectiveness in one or more modeling domains. The lectures begin with the common foundations of state machine and state transition system model, abstraction methods and representations which span multiple model/property types. Queuing network and Petri net models are positioned as endpoints of system abstraction where the only functional behavior retained is macro-level control flow. Finite state machine/state transition models are presented as a general foundation for many types of models. Markov systems are presented as special forms of state transition systems. Formulation of queuing models as Markov systems for evaluation is covered. Functional modeling representations such as Kripke structures and evaluations methods of model checking are sketched. Stochastic process algebras and model checking with metric temporal logics are used to illustrate integration of performance and correctness modeling. Applications of each type of model/property are given with an emphasis on performance and performability.

Methods for model evaluation are also covered. Fundamentals of sequential and parallel/distributed simulation are covered. Methods for analytical evaluation of special forms such as queuing networks are derived from the semantics of the model representations.

The course targets first and second year graduate students. It was offered regularly every other spring semesters in the Computer Sciences Department at UT-Austin through 2005. The workload of the class is a set of assignments focusing on exploration of the commonalities among model/property types and a semester long project where students are encouraged to use methods and tools which apply or enable application of multiple model/property types.

Brief Biography

Browne is Professor of Computer Science and Physics and holds the Regents Chair #2 in Computer Sciences at The University of Texas at Austin. Browne earned his Ph.D. in Chemical Physics at The University of Texas in 1960. He taught in the Physics Department at The University of Texas from 1960 through 1964. He was, from 1965 through 1968, Professor of Computer Science at Queens University in Belfast and directed the Computer Laboratory. Browne rejoined The University of Texas in 1968 as Professor of Physics and Computer Science. He served as Department Chair for Computer Science in 1968-69, 1971-75 and 1984-87.

He founded Hyperformix, an Austin Texas based software company which sells products and services for performance management and engineering for enterprise level computer systems.

He is a Fellow of the Association for Computing Machinery, of the British Computer Society, the American Physical Society, the American Association for the Advancement of Science and the Institute for Constructive Capitalism. He was Chairman of the ACM Special Interest Group on Operating Systems 1973-75 and has been in the past an Associate Editor of several journals. Browne has published approximately 100 papers in computational physics and 250 papers in Computer Science and has supervised the Ph.D dissertation research of 69 students.