Compositional Performance Analysis using Probabilistic I/O Automata*

Eugene W. Stark**

Department of Computer Science, State University of New York at Stony Brook, Stony Brook, NY 11794 USA

In this talk, I give an overview of some recent work, by my colleagues and me at Stony Brook, concerning compositional specification and performance analysis techniques for finite-state reactive systems in which probability and timing play a significant role.

By *reactive systems*, I mean concurrent systems whose components maintain an ongoing interaction with their environments. The complexity of such systems typically resides not in the calculations to be performed by individual system components, but rather in timing and other details of the interactions between components. Examples of reactive systems incude process control software, telecommunication security and e-commerce protocols, and embedded systems. By *compositional specifications* I mean those in which complex systems are described in a hierarchical and modular way as compositions of simpler components. By *compositional performance analysis* I mean performance analysis techniques that exploit the modular structure of a system description and work in a component-by-component fashion to calculate performance measures.

In our work [WSS97,SS98,SP99], we have introduced *Probabilistic I/O Automata* (PIOA) as a model for reactive systems in which probability and timing are of importance. The PIOA model is a probabilistic adaptation of the I/O automata model developed by Nancy Lynch and her students [Lyn96]. A key feature of the I/O automata model is the operation of *composition*, by which a "compatible" collection of I/O automata can be combined into a single, larger automaton. The notion of composition depends in an essential way on a distinction made in the I/O automata model between *input actions*, which are stimuli applied to an automaton by its environment, *output actions*, which are responses made by an automaton to its environment, and *internal actions*, which represent internal steps in which the automaton does not interact with its environment. Output and internal actions are called *locally controlled*, because their occurrence is under the control of the automaton, whereas input actions are under the control of the environment, with the automaton unable to exert any influence over their occurrence.

The PIOA model integrates probability and timing into the I/O automata model, while carrying over in a natural way its essential features of asynchrony and compositionality. To the original I/O automata model, two kinds of probability-related information are added. First, probability distributions are

* Research supported in part by AFOSR Grant F49620-96-1-0087.

** Author's E-mail address: `stark@cs.sunysb.edu`

associated with each state q of an automaton, in such a way that for each input action a, there is one probability distribution covering all transitions for action a from state q, and such that there is one additional probability distribution covering all transitions for locally controlled actions from state q. The second type of probabilistic information included in the PIOA model consists of a *rate* associated with each state. The rate is a nonnegative real number, which we interpret as the parameter of an exponentially distributed random variable that describes the amount of time spent by an automaton in a state before it executes its next locally controlled action. Rates enable us to "probabilize" the scheduling of locally controlled transitions taken by a system of component PIOAs, according to the following *race criterion*: upon entering a state, each component PIOA chooses randomly, according to the exponential distribution whose parameter is the rate of that state, a nonnegative real number that represents the amount of time that PIOA will remain in this state before executing the next locally controlled action. The times chosen by each of the component PIOAs are compared, the PIOA that has chosen the smallest time is declared "the winner," and it is allowed to perform the next locally controlled action at the time it has chosen.

A PIOA with an empty set of input actions is called *closed*. Closed PIOAs determine continuous-time Markov chains (CTMCs) [How71], which are widely used in modeling and performance analysis. An important reason for the widespread use of CTMCs is that they can, in fact, be analyzed—algorithms exist to "solve" the CTMC to compute performance measures that predict the behavior of the real system. For example, the *steady-state probabilities* for a CTMC describe the fraction of time spent by the system in each state over the long run, and these probabilities can be obtained by solving a system of linear "balance equations" associated with the CTMC. From the steady-state probabilities, one can compute the *mean recurrence times*, which describe, on the average, how long it will take for a system to return to a state it has just left. The reciprocals of the mean recurrence times can be interpreted as *throughputs*, which are often useful performance parameters that can actually be measured for a real system.

The PIOA model is closely related to stochastic process algebras such as EMPA [BDG98], PEPA [Hil96], and TIPP [HR94], whose stochastic semantics are also described in terms of an underlying CTMC. Significant differences between the PIOA model and these other stochastic process algebras are: (1) the PIOA model does not attempt to treat any form of non-probabilistic internal choice, and (2) in the PIOA model, a syntactic distinction is drawn between output ("active") actions and input ("passive") actions, and at most one component can be active in any synchronized action. These restrictions enable the PIOA model to avoid semantic problems associated with assigning probabilities to synchronized actions.

Traditional techniques [Ste94] for performance analysis of a CTMC system model typically proceed by first compiling the system description into a matrix representation of a CTMC. An iterative numerical method, such as Gauss-Seidel iteration or successive overrelaxation, is then used to solve the system to obtain, for example, the steady-state probabilities. Although the matrices arising in practice from CTMC descriptions are generally sparse, and require storage space which is linear in the number of states of the CTMC, if no steps are taken to reduce the size of the CTMC representation, the amount of memory needed to store it limits the maximum size of models that can be solved. Typical maximum sizes of CTMCs that can be treated by standard methods are somewhat in excess of 10^6 states. This sounds like a lot of states, until one realizes that it only takes six components, with each component having ten states, to describe a system with this many global states. This severely limits the applicability of standard techniques to even relatively simple real-world systems.

The PIOA-based techniques we have developed can be used to give a description of a CTMC as the composition of a collection of PIOA components, and then to exploit the compositional structure of the system description in the calculation of performance measures, thereby avoiding the construction of the full CTMC state space. Our techniques are applied by starting with a representation of a particular performance measure to be calculated, and then treating the system one component at a time. As each component is treated, state-space reduction is performed before proceeding to the next component. This reduction step eliminates information that is not relevant to the computation of the performance measure of interest. The overall idea is similar to the compositional aggregation technique of Hillston [Hil95], except that we use a very different state-space reduction algorithm and we are able to use information about the performance measure to enhance the reduction.

Our techniques most naturally apply to the calculation of transient performance measures that can be expressed as expectations of a certain kind of function over the probability space of system executions. Some examples of performance measures that can be expressed in this way are: (1) the probability of a total failure of all components, in a system in which components fail and are repaired after some time; (2) the expected time until a total failure in such a system, assuming that such a failure will occur with probability one. We have also shown that our techniques can be applied to compute steady-state performance measures, by viewing them as limits of transient measures.

Our methods for compositional analysis of systems of PIOAs are based on a kind of abstract semantics for PIOAs that retains just the information required for performance analysis. The main notions in the theory are: *rated traces*, which are alternating sequences of rates and actions that form a kind of abstraction of PIOA executions; *observables*, which are measurable functions from rated traces to real numbers, and whose expectations correspond to performance measures; and *PIOA behaviors*, which are functions from observables to observables that capture the way in which system performance is modified by the incorporation of an additional PIOA as a system component. A *compositionality theorem* states that PIOA behaviors compose: the behavior of a composite system can be obtained by composing (as functions) the behaviors of the individual system components.

Our algorithms for compositional performance analysis work for *representable observables*, which are those that have a *linear representation* as a certain kind of automaton with states in a finite-dimensional vector space. Linear representations of observables are a generalization of the classical notion of linear representations for formal power series [BR84]. We have shown [SS98] that algorithms exist for: (1) calculating the result of applying the behavior of a finite-state PIOA to a representable observable; (2) finding, given a linear representation of an observable, a minimum-dimension linear representation for that same observable; and (3) calculating the expectation of an observable to obtain a performance measure. We have implemented our algorithms in the very high-level functional programming language Standard ML. A description of the implementation and the results of applying it to some examples can be found in [SP99].

References

[BDG98] M. Bernardo, L. Donatiello, and R. Gorrieri. A formal approach to the integration of performance aspects in the modeling and analysis of concurrent systems. *Information and Computation*, 144:83–154, August 1998.

[BR84] J. Berstel and C. Reutenauer. *Rational Series and Their Languages*, volume 12 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, 1984.

[Hil95] J. Hillston. Compositional Markovian modelling using a process algebra. In W. Stewart, editor, *Proceedings of the Second International Workshop on Numerical Solution of Markov Chains: Computations with Markov Chains*, Raleigh, North Carolina, January 1995. Kluwer Academic Press.

[Hil96] J. Hillston. *A Compositional Approach to Performance Modelling*. Cambridge University Press, 1996.

[How71] R. A. Howard. *Dynamic Probabilistic Systems, Volume II: Semi-Markov and Decision Processes*, volume 2 of *Series in Decision and Control*. John Wiley & Sons, 1971.

[HR94] H. Hermanns and M. Rettelbach. Syntax, semantics, equivalences, and axioms for MTIPP. In U. Herzog and M. Rettlebach, editors, *Proc. of 2nd Workshop on Process Algebras and Performance Modelling*, Erlangen-Regensberg, Germany, July 1994. Universität Erlangen.

[Lyn96] N. A. Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1996.

[SP99] E. Stark and G. Pemmasani. Implementation of a compositional performance analysis algorithm for probabilistic I/O automata. In *Proceedings of 1999 Workshop on Process Algebra and Performance Modeling (PAPM99)*. Prensas Universitarias de Zaragoza, September 1999.

[SS98] E. W. Stark and S. Smolka. Compositional analysis of expected delays in networks of probabilistic I/O automata. In *Proc. 13th Annual Symposium on Logic in Computer Science*, pages 466–477, Indianapolis, IN, June 1998. IEEE Computer Society Press.

[Ste94] W. J. Stewart. *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, 1994.

[WSS97] S.-H. Wu, S. A. Smolka, and E. W. Stark. Composition and behaviors of probabilistic I/O automata. *Theoretical Computer Science*, 176(1-2):1–38, 1997.