A Construction and Analysis Tool Based on the Stochastic Process Algebra TIPP

Holger Hermanns *
Vassilis Mertsiotakis *
Michael Rettelbach *

ABSTRACT [1] There are many ways to incorporate a notion of time into process algebras in order to integrate functional design and performance analysis. One major research strand, *stochastic* process algebras, concentrates on the annotation of actions with exponentially distributed random variables. This paper presents a tool for the functional analysis and performance evaluation of complex systems based on the stochastic process algebra paradigm. The *TIPP-tool* provides facilities for model specification, reachability analysis, as well as several numerical algorithms for the solution of the underlying Markov chain and the computation of performance measures.

1 Introduction

Stochastic process algebras (SPA) have been introduced as an extension of classical process algebras, like CCS or CSP, with timing information aiming mainly at the integration of functional design and quantitative analysis of computer systems. Time is represented by attaching random variables to every activity in the model, determining their durations [2].

As in classical process algebras, the main characteristic of SPAs is constructivity, i.e. the ability to describe complex systems as a composition of several smaller ones. This concept, together with a powerful abstraction mechanism to hide internal structures, forms the basis of the successful application of SPA for describing and analyzing systems of various domains. The formal foundations of the stochastic process algebra TIPP (*timed processes and performability evaluation*) are presented in [4] and have recently been extended in [5, 7]. They include a formal semantics and an equational theory for congruences that can be seen as stochastic

*Universität Erlangen-Nürnberg, IMMD VII, Martensstr. 3, D–91058 Erlangen

[1] This work is supported in part by the Deutsche Forschungsgemeinschaft (*SFB 182*) and by the BRA ESPRIT project 7269 (*QMIPS*).

counterparts of Milner's strong bisimulation and observational congruence.

The TIPP-tool is a prototype modelling tool supporting a LOTOS-oriented input language. Apart from rudimentary facilities to apply functional analysis, it comes up with a set of numerical solution modules for the stationary as well as transient analysis of the *continuous time Markov chain* (CTMC) underlying a TIPP-specification. The tool has been successfully applied to model and analyze various types of parallel and distributed systems, including an Email-system, the alternating bit protocol, a robot control system, and various multiprocessor systems, see [2] for an overview.

2 The language TIPP

The input language of the tool is closely related to LOTOS, but enriched with stochastic timing information. An exponentially distributed random variable is associated with every activity specifying its duration. Hence, every term in our language can be interpreted as a high level description of a CTMC. Exponential distributions can be characterized by a single parameter. Therefore, each activity of the system possesses a specific name and a parameter, called rate, characterizing its duration.

A specification consists of a number of process definitions, each describing a process scheme with generic activity names and activity rates within a behaviour expression. Various constructs are available to describe behaviour expressions, like the prefix operator (;), choice (□), the halt process (`stop`), the hiding operator (`hide`), or the parallel operator |{*action list*}|.

Additionally, process instantiations may occur within behaviour expressions and resemble the invocation of procedures in procedural programming languages such as PASCAL. A process instance is generated by replacing the list of formal action names and action rates by actual action names/rates in the corresponding process definition. A typical specification clearly resembles LOTOS [1]:

```
behaviour System :=
   hide a in P[a](lambda) |{a}| Q[a,b,c](lambda,mu,1)
where
   process P[in](lambda) := (in,lambda);stop
   endproc
   process Q[ready,wait,msg](xi,mu,nu) :=
            (wait,xi);(msg,mu);Q[ready,wait,msg](xi,mu,nu)
            []
            (ready,nu);stop
   endproc
endspec
```

The behaviour of a specification is defined by a structural operational semantics (see [2] for more details) that maps a specification onto its semantic model, a *Markovian labelled transition system* (MLTS) consisting of nodes representing process terms and arcs between them. These arcs are labelled with pairs of actions and rates. Figure 1 shows the semantic model of the above specification.

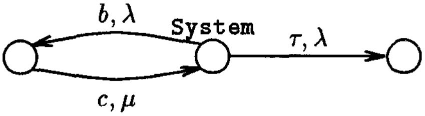

FIGURE 1. Markovian labelled transition system

3 Tool Components

Model descriptions are syntactically and semantically analyzed using a parser written in the functional programming language *Standard ML.* The parser has been generated using the tools *ML-YACC* and *LEXGEN* from the New Jersey ML toolkit. The use of this programming language has shown to be especially useful for implementing the operational semantics. After checking syntactical correctness of a specification the parser applies the operational semantic rules to produce the MLTS. This transition system constitutes the interface between the formal semantics implemented in *ML* and subsequent analysis of the system mostly implemented in *C*. The *C* programming language was chosen in order to allow the efficient solution of very large problems as well as taking advantage of existing libraries for numerical linear algebra [6].

Performance analysis is based on the MLTS where both labels of an arc, denoting the activity name and rate, are regarded as being significant, as are the arc multiplicities. For any SPA model with finite state space the underlying CTMC can be derived directly by associating a state i with each node P of the MLTS. The transitions of the CTMC are the amalgamation of all the arcs joining the nodes, and the transition rate is the sum of the individual activity rates. The activity name information is not incorporated into the CTMC representation but it is often crucial in defining the measures to be extracted from a model.

Characteristic performance and dependability measures based on equilibrium behaviour are extracted via the steady state solution of the CTMC using standard numerical techniques. The resulting values π_i can be interpreted as the probability to observe the behaviour corresponding to each process state i at any time in the steady state. From these values it is straightforward to obtain high-level measures such as throughput, or utilization. The only sufficient condition for a unique equilibrium solution is strong connectivity of the MLTS.

If the latter condition is violated or steady state measures do not give a sufficient insight into the system behaviour, it is possible to carry out transient analysis. Standard algorithms like randomization or calculation of higher moments may be used to find time-related properties of systems. Transient analysis is central to the integration of qualitative and quantitative aspects of model analysis as it allows quantitative implications of qualitative properties of a system to be investigated.

The result of numerical analysis is usually a vector with state probabilities. In order to obtain more sophisticated and expressive results the user can specify measures. This is done via rewards that are assigned to states that match a certain regular expression the user has to specify. This strategy makes use of the fact that the operational semantics associates a distinct process term to every state in the MLTS. Experiment series are also supported by allowing rates to be symbolic variables. The specification of the model as well as the measures and experiments is supported through a graphical user interface. A detailed explanation of the tool and its components can be found in [3].[2]

4 References

[1] T. Bolognesi and E. Brinksma. Introduction to the ISO Specification Language LOTOS. In *The Formal Description Technique LOTOS.* North-Holland, 1989.

[2] N. Götz, H. Hermanns, U. Herzog, V. Mertsiotakis, and M. Rettelbach. Constructive Specification Techniques - Integrating Functional, Performance and Dependability Aspects. In *Quantitative Methods in Parallel Systems.* Springer, 1995.

[3] H. Hermanns and V. Mertsiotakis. A Stochastic Process Algebra Based Modelling Tool. In *Proc. of the 11th U.K. Perf. Eng. Workshop for Computer and Telecommunication Systems.* Springer, 1995.

[4] H. Hermanns and M. Rettelbach. Syntax, Semantics, Equivalences, and Axioms for MTIPP. In *Proc. of the 2nd Workshop on Process Algebras and Performance Modelling.* IMMD, Universität Erlangen, 1994.

[5] H. Hermanns, M. Rettelbach, and T. Weiß. Formal Characterisation of Immediate Actions in SPA with Nondeterministic Branching. *The Computer Journal*, 38(7), December 1995. Special issue: Proc. of the 3rd Workshop on Process Algebras and Performance Modelling.

[6] S.K. Kundert and A. Sangiovanni-Vincentelli. A Sparse Linear Equation Solver. Technical report, University of California, Berkeley, 1988.

[7] M. Rettelbach. Probabilistic Branching in Markovian Process Algebras. *The Computer Journal*, 38(7), December 1995. Special issue.

[2] More information about the TIPP project are available at the URL `http://www7.informatik.uni-erlangen.de/tipp/`