Integrated Analysis of Concurrent Distributed Systems using Markovian Process Algebra *

Marco Bernardo[a] and Lorenzo Donatiello[a] and Roberto Gorrieri[a]

[a]Dipartimento di Matematica, Università di Bologna
Piazza di Porta S. Donato 5, 40127 Bologna, Italy

Since the 60's a lot of work has been done in order to devise modeling techniques which are adequate to represent concurrent systems. Process algebras are one of the most relevant results of this work. As a matter of fact, (i) process algebras are *abstract languages* conceived for defining formally concurrent systems in a *compositional* way; i.e., they provide the designer with a set of operators by means of which constructing complex systems from simpler ones; hence, the designer is led to use a hierarchical and modular design style. And (ii) with process algebras concurrent systems can be given formal descriptions which are more easily understandable and more easily modifiable than the ones obtained by many other models.

Classical process algebras, like LOTOS [2], can be used to describe only the *functional aspect* of the behavior of concurrent systems; neglecting their *temporal aspect* is a remarkable drawback for the expressiveness of classical process algebras because (i) time-critical concurrent systems (e.g. real time systems) cannot be modeled in a completely satisfactory manner; moreover, (ii) the performance of the concurrent systems cannot be estimated. It often happens that a concurrent system is firstly fully designed and tested for functionality and afterwards tested for efficiency; so, if the performance is detected to be poor, the concurrent system has to be redesigned with negative consequences for both the design costs and the delivery at a fixed deadline.

The need of integrating the performance analysis of a concurrent system into the design process of the system itself has recently stimulated many researchers. In the framework of process algebras, this need is satisfied by introducing the concept of time to express the durations of the activities performed by the concurrent systems being modeled; when such durations are expressed in a deterministic way, the process algebra is said to be a *temporal process algebra* (see, e.g., [6]), whereas when such durations are expressed through random variables, it is said to be a *stochastic process algebra* [7, 8].

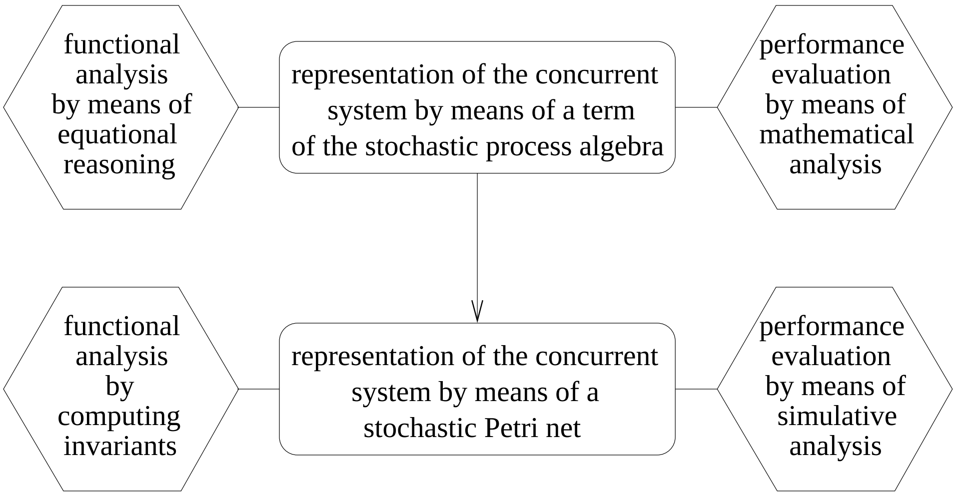
In this short abstract we introduce a few ideas underlying a new stochastic process algebra, named *MPA (Markovian Process Algebra)*, which is not actually introduced here for lack of space. Indeed, the purpose of this note is not that of formally defining MPA, rather that of giving an idea about what can be done with MPA. The reader interested in the formal definitions is invited to consult the full version of this paper [1] and the three technical reports (TR-94-10, 94-11 and 94-12) by the same authors.[1]

*This reaserch has been partially funded by CNR and MURST.

[1]The technical reports are available via anonymous FTP from the area `ftp.cs.unibo.it:/pub/TR/UBLCS` in compressed PostScript format.

MPA is supplied with an operational interleaving semantics, a markovian semantics and an operational net semantics. Therefore, there are three semantic models associated with each MPA term: an *interleaving model* represented by a labeled transition system, a *stochastic model* represented by the state transition rate diagram of a homogeneous continuous time Markov chain (HCTMC), and a *distributed model* represented by a generalized stochastic Petri net (GSPN) [9].

With a stochastic process algebra like MPA we can exploit a modeling technique for concurrent systems which integrates different views of concurrent systems (*centralized* vs. *distributed*) as well as the different aspects of their behaviour (*qualitative-functional vs. quantitative-temporal*) of concurrent systems. Such a modeling technique can be summarized by the following scheme



which points out the following integrated specification strategy:

- The first phase consists of specifying the concurrent system as a term of MPA, so as to obtain a first representation, easy to understand and compositional. With this algebraic representation, it is possible to:

  - Perform a functional analysis of the concurrent system by, e.g., equivalence checking or, if the equivalence relation is a congruence, by equational reasoning defined on its axiomatization. Such analyses can detect qualitative properties of the concurrent system (e.g. deadlock-freedom) and can also help in minimizing the state space of the system representation. These analyses can be also computer-aided (see, e.g., [5]).
  - Evaluate the performance of the concurrent system by resorting to the study of its associated HCTMC, which can be assisted by computer (e.g., [11]).

- The second phase consists of translating the term of the stochastic process algebra into a generalized stochastic Petri net (GSPN), giving a distributed representation of the same concurrent system. Such a translation makes explicit the parallelism and the causal dependencies among the activities of the concurrent system itself, but the price to pay is that such a model usually has a remarkable graphic complexity. Moreover, it is not easily compositional, hence making hard, in general, the detection and the analysis of its subsystems. With this stochastic Petri net representation, it is possible to:

- Perform a qualitative analysis of the concurrent system by, e.g., computing net invariants [3].
- Evaluate the performance of the concurrent systems by resorting to existing tools for simulation on stochastic Petri nets, e.g. [4].

As the various (transformations producing the) semantics we have proposed for MPA can be fully mechanized, future work will be devoted to implement a software tool which associates to each MPA term the collection of its three semantics. On this basis, the next step should consist of integrating this tool with the various ones, already available, tailored for specific purposes. For instance, the interleaving semantics is a labelled transition system which can be input for AUTO [10]; the Markov chain associated to a term can be given as input to SHARPE [11]; its Petri net to PAPETRI [3] and its GSPN to GreatSPN [4]. Therefore, our work opens a perspective to a fully integrated tool which can help in making computer aided – and possibly automatic – the qualitative and quantitative analysis of concurrent distributed systems.

REFERENCES

1. M. Bernardo, L. Donatiello, R. Gorrieri, "Integrating Performance and Functional Analyses of Concurrent Systems with MPA", Technical Report, July 1994.
2. T. Bolognesi, E. Brinksma, "Introduction to the ISO Specification Language LOTOS", *Computer Networks and ISDN Systems*, vol. 14, 1987.
3. G. Berthelot, C. Johnen, L. Petrucci, "PAPETRI: Environment for the Analysis of Petri Nets", in Proc. CAV'90, LNCS 531, Springer, 1991.
4. G. Chiola, "GreatSPN 1.5 Software Architecture", in Proc. of the *Fifth Int. Conf. on Modeling Techniques and Tools for Computer Perf. Eval.*, Turin, 1991.
5. R. Cleaveland, J. Parrow, B. Steffen, "The Concurrency Workbench: A Semantics-Based Tool for the Verification of Concurrent Systems", in ACM *Trans. on Programming Languages and Systems* 15, 1993.
6. R. Gorrieri, M. Roccetti, E. Stancampiano "A Theory of Processes with Durational Actions", to appear in *Theoretical Computer Science*, also available as TR-93-27, Università di Bologna, 1993.
7. N. Götz, U. Herzog, M. Rettelbach, "Multiprocessor and Distributed System Design: the Integration of Functional Specification and Performance Analysis Using Stochastic Process Algebras", in LNCS 729, Springer, 1993.
8. J. Hillston, "A Compositional Approach to Performance Modelling", Ph.D. thesis, CST-107-94, University of Edinburgh (UK), 1994.
9. M. A. Marsan, "Stochastic Petri Nets: an Elementary Introduction", in LNCS, vol. 424, Springer-Verlag, 1990.
10. V. Roy, R. de Simone, "Auto / Autograph", in Proc. CAV'90, LNCS 531, Springer, 1991.
11. R.A. Sahner, K.S. Trivedi, "Reliability Modelling Using SHARPE", *IEEE Trans. Reliabil.* 13 (2), 186-193, 1987.