A Compact Semantic Model for Characterization of Stochastic Temporal Properties of Concurrent Systems

Mokdad Arous [1], Jean-Michel Ilié [2] And Djamel-Eddine Saïdouni [1]

[1] **MISC Laboratory, Computer Science Department, University of Mentouri Constantine, 25000, Algeria**

[2] **LIP6 Laboratory, Computer Science Department, Paris 6 University Paris, 75005, France**

Abstract

In this paper, a new semantic model is proposed for characterizing the performance properties of stochastic concurrent systems, called *Maximality-based Labeled Stochastic Transition System* (MLSTS). A Stochastic Process Algebra, called S-LOTOS, is associated with these MLSTS models as a specification language to describe the stochastic temporal aspects of concurrent systems under the assumption of generally distributed durations of actions. The MLSTS models can be automatically generated from S-LOTOS specifications according to the (true concurrency) maximality semantics. In addition, we bring out the main advantage of the MLSTS as it is shown on practical examples in reducing the number of states and transitions w.r.t. standard ST-semantic models, which are frequently used in specification and modeling of stochastic systems with non-Markovian Process Algebra. Using our MoVeS Tool, we bring out the main advantage of our approach which is a drastic reduction obtained in the number of states and transitions, in comparison to standard ST-semantic models.

Keywords: *Maximality Semantics, Semantic Models, ST-Semantics, Stochastic (non-Markovian) Process Algebra, Stochastic Transition Systems.*

1. Introduction

Nowadays, parallel and distributed systems have become the foundations of many application areas. However the correctness and the performances of the proposed constructions both remain difficult to prove since models of systems do not easily deal with all the necessary concrete parameters together, like competition, random phenomena, synchronization and non-determinism. Hence, the specification of an exact timing concerning the expected behaviors may lack of consistency. Often, good models come from considering the behaviors related to random timing. In transmission systems for instance, transmission errors and decision changes in the traffic flow produced randomly, lead to various communication delays. In fact, there is a need of adequate stochastic timing models, for the specification and verification of such stochastic behaviors.

Several models were already developed to capture randomly varying time instants and also time intervals, among with Queuing models, e.g. [21], and stochastic versions of Petri Nets, e.g. [5, 7], Automata, e.g. [18], and Process Algebras, e.g. [1, 8, 9, 17]. Among the specification languages, Stochastic Process Algebras (SPAs) take advantage from both compositionality and abstraction aspects (i.e. build up of complex models from detailed components and their interactions, but disregarding internal behavior when it is appropriate to do so), whereas providing a formal description context.

Two main SPA approaches have been adopted for expressing random time properties of stochastic systems. The Markovian Process Algebras (MPAs) accord with the interleaving semantics [1, 9, 13]. The specification of the durations of actions is limited to exponential distributions, hence MPAs take advantage from the memoryless property of exponential distributions, which yield analytically tractable models in the form of Continuous Time Markov Chains (CTMCs) [2, 22]. However, exponential distribution laws appear to be a limitation in expressiveness for many concrete situations, for instance, in case of only the minimum and maximum of some quantity are known, the uniform distribution would be a better law to consider. Face with this limitation, non-Markovian Process Algebras are proposed. These second SPA family adopts general probability laws to specify action durations.

Contrary to exponential distributions, general distributions allow handling the residual durations of running actions. Therefore, the interleaving semantics is no more appropriate [10]. Instead, the specifications of generally distributed durations refer to the true concurrency semantics. Thus, the system behaviors are never more represented like totally ordered sequences like in the interleaving semantics, but adequately like partial order ones, allowing one to consider non-atomic actions.

The existing non-Markovian SPA approaches introduce an explicit representation of the start and end events for every running actions. This allows considering a specific true concurrency semantics called ST-Semantics (for *Start* and *Termination*). For instance, IGSMP (*Interactive Generalized Semi-Markov Process*) [14, 16], GSMPA (*Generalized Semi-Markovian Process Algebra*) [15, 16], SPADES (*Stochastic Process Algebra for Discrete Event Simulation*) [10, 17], are models which adopt both the ST-Semantics and general distribution laws.

Anyway, these approaches suffer from an increasing exponential blow up implied by the combinations of the start and termination events. Alternatively, we refer to a more appropriate semantics, namely maximality semantics and defined in [3, 4] for qualitative needs. our approach aims at handling true concurrency notions, for specifying non-Markovian properties, without being attacked by the state space explosion problem inherent to the splitting of actions.

In this paper, we formally introduce the *Maximality-based Labeled Stochastic Transition System* (MLSTS), as a new semantic model for the characterization of performance properties of stochastic concurrent systems. This follows a first work [11] which show that the maximality based semantic models describe the same qualitative and quantitative properties as specified in the ST-Semantic models. In this paper, we also present our Stochastic Process Algebra language S-LOTOS, which allows specification of stochastic temporal aspects of concurrent systems, as a language to describe MLSTSs compactly. From S-LOTOS specification, we show how MLSTSs can be generated automatically, according to the maximality semantics.

The paper is scheduled as follows: In section 2, the main principles and formal definition of the MLSTS models are presented. In section 3, we present our Algebraic language S-LOTOS, then we define the derivation rules useful to generate MLSTS models from S-LOTOS specifications. Next, in section 4, the generation of MLSTS is briefly discussed, and the main features of our MoVeS tool are exhibited, and experimental results show that our MLSTS models are very reduced w.r.t standard ST-semantic models. Section 5 concludes the paper and opens some perspectives.

2. Maximality-based Labeled Stochastic Transition Systems

The MLSTS models are defined as state/transition systems. Unlike ST semantic models, each transition only represents the start of an action execution. Since actions are not considered as atomic, the concurrent execution of multiple actions can be represented, and allows the distinction between sequential and parallel executions.

In MLSTS, the running actions are specified at the state level. Each instance of running actions is called a *maximal event* and is identified by a distinct name. In fact, each state of the system is featured by a unique configuration [3]. The configuration of a state s is denoted ${}_M[E]$ s.t. M is the set of maximal events in s and E is the behavior expression considered from s. Every transition defined from s is labeled by ${}_C(a, f)_x$ expressing that a is an action that can be activated from E in case the maximal events of the subset $C \subseteq M$ are terminated. Further, C is called the *causality set* of the transition. The symbol x is the name identifying the *start event* of the new execution of a. The event identification is required to avoid confusion because several instances of running actions can have the same action name.

A detailed presentation of the maximality semantics can be found in [4]. Let us now illustrate the principles of the maximality semantics used in the MLSTS models through the two systems presented in Figure 1, modeled in Petri Nets and both based on two actions a and b with probability distribution functions f and g respectively. In the system E, a is executed in parallel with b, whereas *in F*, either the execution of a is followed by the one of b or *vice versa*. The corresponding MLSTSs representing the behaviors of E and F, obtained by applying the maximality semantics, are represented in Figure 2. As an initial situation, there is no action in execution, therefore the sets of maximal events attached to the initial states of E and F are empty. As a consequence, the initial configurations of E and F are ${}_\varnothing[E]$ and ${}_\varnothing[F]$, respectively. By assuming that the action a happens first from E and F, the corresponding transitions are represented as:

$$\varnothing\, E1 \xrightarrow{{}_\varnothing(a,f)_x} {}_{\{x\}} E2$$

$$\varnothing\, F1 \xrightarrow{{}_\varnothing(a,f)_x} {}_{\{x\}} F2$$

where x is the event name identifying the starting of a. In both resulting states $E2$ and $F2$, x is a maximal event.

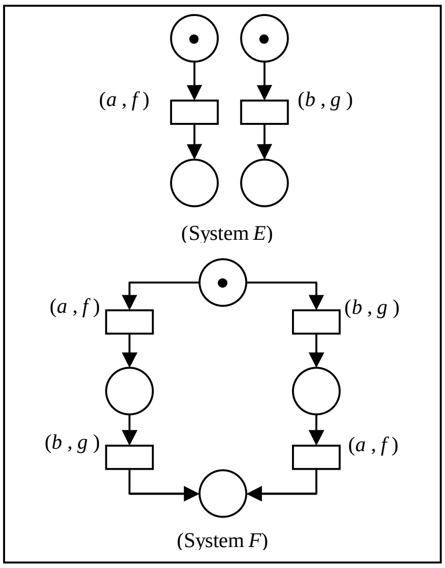


Fig. 1. Example of two systems *E* and *F*

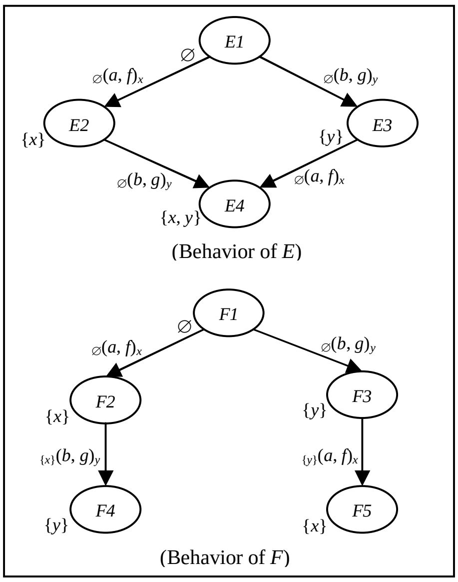


Fig. 2. Behaviors of *E* and *F* according to the maximality semantics

From the state *E2*, the following transition occurs in case *b* starts:

$$_{\{x\}}E2 \xrightarrow{\varnothing(b,g)_y} {}_{\{x,y\}}E4$$

Here, *y* is the event name identifying the start of *b*, which does not depend on the termination of *a* due to the parallel execution semantics considered.

From the state *F2*, because of the sequential execution of actions *a* and *b*, we deduce that the start of *b* is constrained by the causality dependence against *x*. Actually, it is submitted to the end of the execution of *a*. This results in the following transition:

$$_{\{x\}}F2 \xrightarrow{\{x\}(b,g)_y} {}_{\{y\}}F4$$

Regarding the resulting state *F4*, the only maximal event is the one identified by *y*, representing the start of execution of *b*. This state *E4* has a different situation : two maximal events appear (identified by *x* and *y*). Observe that a symmetric scenario could happen whenever the action *b* occurs first (see Figure 2).

2.1 Formal Definition of MLSTS

An MLSTS is defined as follows:

Definition 1. Maximality-based Labeled Stochastic Transition System (MLSTS).
Let $\mathcal{M}$ be a countable set of event names.
An MLSTS is a structure (Ω, *A*, *DF*, *L*, μ, ξ, ψ) with:

- $\Omega = (S, s_0, T, \alpha, \beta)$: is a Transition System s.t. *S* is the countable set of states for the system, at least including the initial state s_0. *T* is the countable set of transitions specifying the states changes. α, β are two mappings $T \rightarrow S$. respectively associating its source $\alpha(t)$ and its target $\beta(t)$ with every transition.
- *A* : is a (finite) set of actions
- *DF* : is a finite set of probability distribution functions ($\Re \rightarrow [0, 1]$).
- $L: T \rightarrow (A \times DF)$ associates with each transition, a pair composed of an action and a probability distribution function. Further, (Ω, ($A \times DF$)) is a transition system labeled over the alphabet ($A \times DF$).
- $\psi : S \rightarrow 2^{\mathcal{M}}_{fn}$ associates with each state, a finite set of maximal event names (in the state).
- $\mu : T \rightarrow 2^{\mathcal{M}}_{fn}$ associates with each transition, a finite set of maximal event names of actions, such that their terminations globally allow the start of this transition. This set corresponds to the direct causes of the transition.
- $\xi : T \rightarrow \mathcal{M}$ associates with each transition, an event name used to identify a start execution of the corresponding action. For any transition $t \in T$, we have:

$\mu(t) \subseteq \psi(\alpha(t))$
$\xi(t) \notin \psi(\alpha(t)) - \mu(t)$
$\psi(\beta(t)) = (\psi(\alpha(t)) - \mu(t)) \cup \{\xi(t)\}$ ♦

From an MSLTS, one can derive two semantic models; a functional one enhancing true concurrency behaviors and a performance one. The functional model is obtained by abstracting the quantitative information related to the various durations of actions, whereas the performance model is obtained by abstracting the functional information. Let us now focus on the performance one.

2.2 The Underlying Performance Models of MLSTS

In probability theory, systems able to execute concurrent activities with general distributed durations are represented by Generalized Semi-Markov Processes (GSMPs) [14, 15]. In [19], GSMPs are introduced as stochastic processes for the modeling of complex phenomena. They generalize CTMCs by allowing the action durations to be submitted to a non exponential distribution, thus timing not only depends on the current state but also on the past of the state. As far as MLSTS are concerned, one can derive a functional model, (by removing the quantitative information related to the durations of actions), and a performance model (by abstraction of the functional information). The performance model of an MLSTS (i.e. its underlying stochastic process) model is a GSMP [12].

3. S-LOTOS as a language to describe MLSTS

In this section, we briefly recall the syntax and semantics of the behavioral expressions of our SPA called S-LOTOS [12]. Based on these expressions, we define S-LOTOS as a language to describe MLSTS. S-LOTOS deals with general probability distributions instead of restricting to exponential ones. A tool for generating automatically MLSTSs from algebraic specifications with S-LOTOS according the maximality semantics is developed and presented here. The reader is assumed to be familiar with the syntax of Basic LOTOS, a standard process algebra, from which S-LOTOS derives. See [12] for details about the semantics of the different operators.

3.1 S-LOTOS behavioral expressions

Considering some concurrent system, let A be the set of observable actions ranged over by a, b, … and L denote any subset of A. The set of all actions that is finally considered, is denoted by Act ($Act = A \cup \{i, \delta\}$) where $\delta \notin A$ is a particular observable action used to notify the successful termination of processes, and i denotes any internal (unobservable) action. Introduce DF as the set of (continuous) probability distribution functions ($\Re \rightarrow [0, 1]$), ranged over by f, g,… Then, define the set B ranging over by E, F... which are behavior expressions that can specify the studied system, according to the following syntax of expressions:

$$E ::= stop \mid exit \mid (a, f)\,; E \mid (i, f)\,; E \mid E\,[]\,E \mid E\,|[L]|\,E \mid hide\ L\ in\ E \mid E\,[b_1/a_1, \ldots, b_n/a_n]$$

In S-LOTOS, stochastic time is handled by using arbitrary distribution functions. An action is represented by a pair (a, f), where a is the action name and f is the probability distribution function that governs the duration of a.

3.2 Configurations and Derivation Rules

We briefly recall the definition of configurations [3], then an operational semantics is defined from the behavior expressions of S-LOTOS, to derive the possible transitions linking the configurations of a given system, leading to the underlying MLSTS model.

Let $\mathcal{M}$ be the set of event names, ranged over by x, y … Further, the notation $2^{\mathcal{X}}_{fn}$ represents the set of finite subsets of a set $\mathcal{X}$.

Definition 2. Configurations.
The set C of configurations is given by:

$\forall E \in \mathcal{B}, \forall M \in 2^{\mathcal{M}}_{fn} : {}_M[E] \in C$

$\forall P \in PN, \forall M \in 2^{\mathcal{M}}_{fn} : {}_M[P] \in C$

if $\varepsilon \in C$ then : $hide\ L\ in\ \varepsilon \in C$

if $\varepsilon, \mathcal{F} \in C$ then : $\varepsilon\,[]\,\mathcal{F} \in C$, $\varepsilon\,|[L]|\,\mathcal{F} \in C$

if $\varepsilon \in C$ and $\{a_1, \ldots, a_n\}, \{b_1, \ldots, b_n\} \in 2^{\mathcal{A}}_{fn}$,

then: $\varepsilon[b_1/a_1, \ldots, b_n/a_n] \in C$ ♦

The operational semantics of S-LOTOS is summarized in Table 1. Let S be the set of states. Further, the transitions between the states stand for transitions between the configurations of these states. The transition relation between configurations is denoted $\rightarrow$. $\rightarrow \subseteq C \times Atm \times C$, where the set of atoms of support ($Act \times DF$) is $Atm = 2^{\mathcal{M}}_{fn} \times (Act \times DF) \times \mathcal{M}$. For any subset of event names $M \in 2^{\mathcal{M}}_{fn}$, $(a,f) \in (Act \times DF)$ and $x \in \mathcal{M}$, the atom (M, (a, f), x) will be denoted ${}_M(a,f)_x$. The choice of an event name can be realized deterministically by using any function get: $2^{\mathcal{M}} \backslash \{\emptyset\} \rightarrow \mathcal{M}$ satisfying $get(M) \in M$, for all $M \in 2^{\mathcal{M}} \backslash \{\emptyset\}$. Function ψ: $S \rightarrow 2^{\mathcal{M}}_{fn}$ corresponds to the function defined in Definition 1. Lastly, the predicate $Wait$: $2^{\mathcal{M}}_{fn} \rightarrow \{true, false\}$ characterizes the termination of the maximal events s.t.: $Wait(M) = true$ if there is at least one running action referenced in M.
The initial state of the underlying MLSTS corresponding to a given S-LOTOS specification E corresponds to the configuration ${}_{\{\}}[E]$. The set of states are built on the fly, s.t. each one is associated with a configuration reachable

from the initial configuration by using the derivation rules of Table 1.

Table 1. Operational Semantics of S-LOTOS

$$\frac{\neg(wait(M))}{{}_{M}[stop] \xrightarrow{{}_{M}(\delta,0)_x} {}_{\phi}[exit]} \quad x = get(\mathcal{M})$$

$$\frac{\neg(wait(M))}{{}_{M}[(a,f);E] \xrightarrow{{}_{M}(a,f)_x} {}_{\{x\}}[E]} \quad x = get(\mathcal{M}) \qquad\qquad \frac{\neg(wait(M))}{{}_{M}[(i,f);E] \xrightarrow{{}_{M}(i,f)_x} {}_{\{x\}}[E]} \quad x = get(\mathcal{M})$$

$$\frac{\varepsilon \xrightarrow{{}_{M}(a,f)_x} \varepsilon' \quad a \notin L}{\mathcal{F}\,|[L]|\,\varepsilon \xrightarrow{{}_{M}(a,f)_x} \mathcal{F}\,|[L]|\,\varepsilon'} \qquad\qquad \frac{\varepsilon \xrightarrow{{}_{M}(a,f)_x} \varepsilon' \quad a \notin L}{\varepsilon\,|[L]|\,\mathcal{F} \xrightarrow{{}_{M}(a,f)_x} \varepsilon'\,|[L]|\,\mathcal{F}}$$

$$\frac{\varepsilon \xrightarrow{{}_{M}(a,f)_x} \varepsilon' \quad \mathcal{F} \xrightarrow{{}_{N}(a,g)_y} \mathcal{F}' \quad a \in L}{\varepsilon\,|[L]|\,\mathcal{F} \xrightarrow{{}_{M \cup N}(a,h)_z} \varepsilon'[z/x]\,|[L]|\,\mathcal{F}'[z/y]} \quad z = get(\mathcal{M} - ((\psi(\varepsilon) \cup (\psi(\mathcal{F})))$$

$$\frac{\varepsilon \xrightarrow{{}_{M}(a,f)_x} \varepsilon'}{\mathcal{F}[\,]\,\varepsilon \xrightarrow{{}_{M}(a,f)_x} \varepsilon'} \qquad\qquad \frac{\varepsilon \xrightarrow{{}_{M}(a,f)_x} \varepsilon'}{\varepsilon[\,]\mathcal{F} \xrightarrow{{}_{M}(a,f)_x} \varepsilon'}$$

$$\frac{\varepsilon \xrightarrow{{}_{M}(a,f)_x} \varepsilon' \quad a \notin \{a_1, \ldots, a_n\}}{\varepsilon[b_1/a_1, \ldots, b_n/a_n] \xrightarrow{{}_{M}(a,f)_x} \varepsilon'[b_1/a_1, \ldots, b_n/a_n]} \qquad \frac{\varepsilon \xrightarrow{{}_{M}(a,f)_x} \varepsilon' \quad a = a_i \quad 1 \le i \le n}{\varepsilon[b_1/a_1, \ldots, b_n/a_n] \xrightarrow{{}_{M}(b_i,f)_x} \varepsilon'[b_1/a_1, \ldots, b_n/a_n]}$$

$$\frac{\varepsilon \xrightarrow{{}_{M}(a,f)_x} \varepsilon' \quad a \notin L}{hide\,L\,in\,\varepsilon \xrightarrow{{}_{M}(a,f)_x} hide\,L\,in\,\varepsilon'} \qquad\qquad \frac{\varepsilon \xrightarrow{{}_{M}(a,f)_x} \varepsilon' \quad a \in L}{hide\,L\,in\,\varepsilon \xrightarrow{{}_{M}(i,f)_x} hide\,L\,in\,\varepsilon'}$$

4. Implementation and Results

Recently, we have define a syntax for describing full S-LOTOS specification of systems [6]. This is now integrated in our tool, called MoVeS (for environment for ***Mo**deling and **Ve**rifying **S**tochastic **S**ystems*), which is used to specify systems in S-LOTOS specifications and to translate them in their underlying MLSTS models, using the former operational rules. Shortly speaking, a system specification S-LOTOS contains :

- ✓ A header describing for a system, its name and parameters (i.e. actions with their distribution functions).
- ✓ A behavioral specification which accords with the syntax defined in Section 3.1.
- ✓ The definitions of processes composing the system.
- ✓ The specification of each used probability distribution function, in particular its type and effective parameters, with regards to the distribution functions used in the system header.

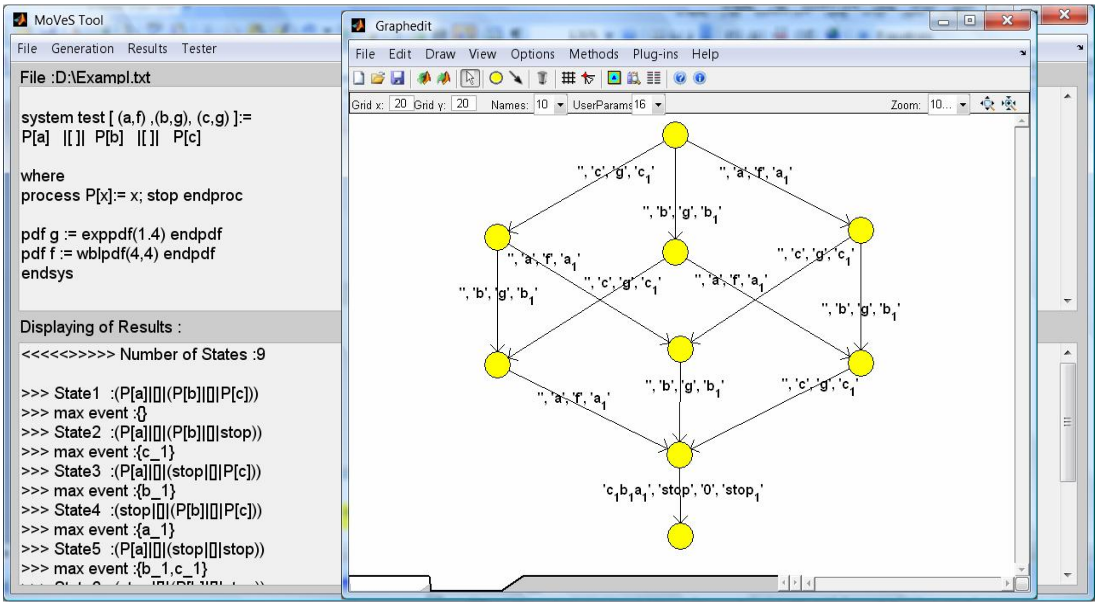


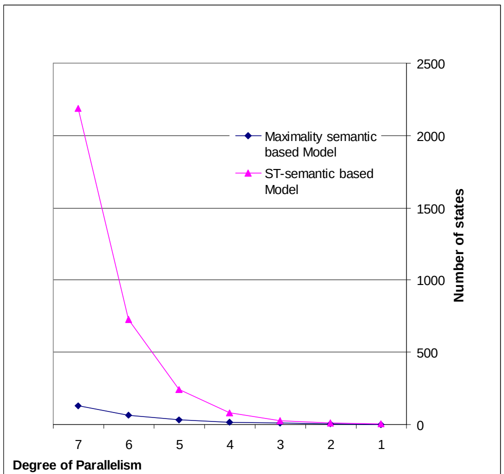
Fig. 3. Example of generating MLSTS models from S-LOTOS specifications using MoVeS tool

The MoVeS Tool user interface is presented in Figure 3, wherein the studied system is obtained by a parallel composition of three processes. The underlying MLSTS models appears textually in the lower-left part of the user interface and graphically in the right part. This graphic visualization is in fact supported by using the rich graphic module Graphedit of TORSCHE Scheduling toolbox [20].

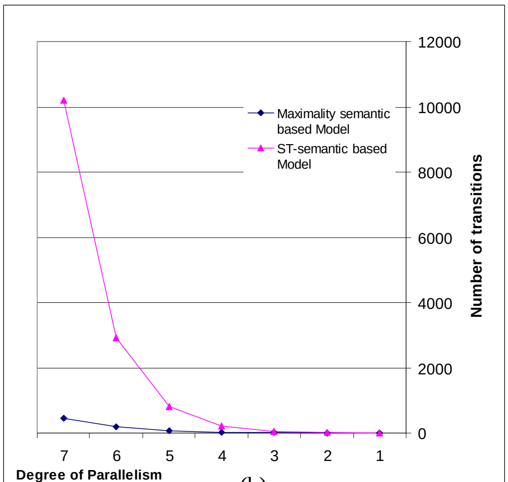
Table 2 brings out the number of states and transitions obtained for different systems having different degrees of parallelism. In reference to the results obtained with the ST-semantic models, the maximality semantics based one become more and more compact, as the parallelism degree increases. Figure 4 graphically demonstrates the reduction according to the considered degrees of parallelism.

Table 2. Comparative Study between MLSTS models and standard ST-semantic models

	Maximality semantics based Model		**ST-semantics based Model**	
Degree of Parallelism	**Number of states**	**Number of Transitions**	**Number of states**	**Number of Transitions**
A simple processes : P[x] = x ; stop	2	3	4	3
Degree of Parallelism = 2 : P[a] |[]| P[b]	5	5	10	13
Degree of Parallelism = 3 (the example in figure 03) : P[a] |[]| P[b] |[]| P[c]	9	13	28	55
Degree of Parallelism = 4 : P[a] |[]| P[b] |[]| P[c] |[]| P[d]	17	33	82	217
Degree of Parallelism = 5 : P[a] |[]| P[b] |[]| P[c] |[]| P[d] |[]| P[e]	33	81	244	811
Degree of Parallelism = 6 : P[a] |[]| P[b] |[]| P[c] |[]| P[d] |[]| P[e] |[]| P[f]	65	193	730	2917
Degree of Parallelism = 7 : P[a] |[]| P[b] |[]| P[c] |[]| P[d] |[]| P[e] |[]| P[f]|[]| P[g]	129	449	2188	10207



(a)



(b)

Fig. 4. The reduction of the numbers of states (a) and transitions (b) in MLSTS models w.r.t standard ST-semantic models.

5. Conclusions and perspectives

The new proposed semantic model, called MLSTS (*Maximality-based Labeled Stochastic Transition System*), well characterizes the stochastic temporal properties of concurrent systems, under the assumption of generally distributed durations. Moreover, the S-LOTOS language appears to be a high level specification for the definition of MLSTS models, which can be considered as a high level formalization of Generalized Semi-Markov Processes (GSMPs).

Our first experimental results are very attractive. A drastic reduction in the numbers of edges and nodes has been highlighted, in reference to the standard ST-semantic models, which are frequently used in specification and modeling of stochastic systems with non-Markovian Process Algebras. We hope that this reduction will help our next steps consisting in integrating in our tool, analysis and verification techniques dedicated to MLSTS models.

References

[1] A. Clark, S. Gilmore, J. Hillston, and M. Tribastone, “Stochastic Process Algebras”, In. M. Bernardo and J. Hillston (Eds.): *Formal Methods for Performance Evaluation*, LNCS 4486, Springer-Verlag, 2007, pp. 132–179.

[2] B. R. Haverkort, “Markovian Models for Performance and Dependability Evaluation”, In. E. Brinksma, H. Hermanns, and J.-P. Katoen (Eds.): *Lectures on Formal Methods and Performance Analysis*, LNCS 2090, Springer-Verlag, 2001, pp. 38–83.

[3] D. E. Saïdouni and J. P. Courtiat, “Prise en compte des durées d’action dans les algèbres de processus par l’utilisation de la sémantique de maximalité”, In. *Proceedings of CFIP’2003. Hermes, France*, 2003.

[4] D. E. Saidouni, “Sémantique de maximalité : Application au raffinement d’actions en LOTOS”, PhD Thesis, LAAS-CNRS, 7 av. du Colonel Roche, 31077 Toulouse Cedex France, 1996.

[5] F. Bause, and P. S. Kritzinger, “Stochastic Petri Nets - An Introduction to the Theory -”, 2nd Edition, Vieweg Verlag, Germany, 2002, ISBN: 3-528-15535-3

[6] Mokdad Arous, “Using the maximality semantics for specifying temporal properties of stochastic concurrent systems”, *Technical report*, MISC laboratory, Mentouri University, Constantine, Algeria, September 2012.

[7] G. Balbo, “Introduction to Stochastic Petri Nets”, In. E. Brinksma, H. Hermanns, and J.-P. Katoen (Eds.): *Lectures on Formal Methods and Performance Analysis*, *Lecture Notes in Computer Science* 2090, Springer-Verlag Berlin Heidelberg, 2001, pp. 84–155.

[8] H. Hermannsa, U. Herzogb, and J.-P. Katoen, “Process algebra for performance evaluation”, *Theoretical Computer Science* 274, Elsevier Science B.V, 2002, pp. 43–87.

[9] J. Hillston, “Process Algebras for Quantitative Analysis”, In. *Proceedings of the 20th Annual IEEE Symposium on Logic in Computer Science (LICS’05)*, 2005, pp. 239-248.

[10] J.-P. Katoen and P.R, D'Argenio, “General Distributions in Process Algebra”, In. E. Brinksma, H. Hermanns, and J.-P. Katoen (Eds.): *Lectures on Formal Methods and Performance Analysis*, LNCS 2090, Springer-Verlag, 2001, pp. 375–429.

[11] M. Arous, D. E. Saidouni and J. M. Ilié, “Addressing State Space Explosion Problem in Performance Evaluation Using Maximality-based Labeled Stochastic Transition Systems”, to appear in *The proceedings of the 2nd International Conference on Computer and Software Modeling - ICCSM* 2012, October 20-21, 2012, Cochin, India. IPCSIT ISSN: 2010-460X.

[12] M. Arous, D. E. Saidouni and J. M. Ilié, “Maximality Semantics based Stochastic Process Algebra for Performance Evaluation”. In. 1*st IEEE International Conference on Communications, Computing and Control Applications (CCCA’11)* March 3-5, 2011 at Hammamet, Tunisia. IEEE Catalog Number: CFP1154M-ART, ISBN: 978-1-4244-9796-6.

[13] M. Bernardo, “Theory and Application of Extended Markovian Process Algebra”, Ph.D. Thesis, University of Bologna (Italy), 1999.

[14] M. Bravetti and R. Gorrieri, “The theory of interactive generalized semi-Markov processes”, In. *Theoretical Computer Science* 282, 2002, pp. 5–32.

[15] M. Bravetti, M. Bernardo, and R. Gorrieri, “Towards Performance Evaluation with General Distributions in Process Algebra”, In. D. Sangiorgi and R. de Simone (Eds.): *LNCS 1466, Proceedings of CONCUR'98, Nice (France), Springer*, 1998, pp. 405-422.

[16] M. Bravetti, “Specification and Analysis of Stochastic Real-time Systems”, PhD thesis, Università di Bologna, Padova, Venezia, 2002.

[17] P. R. D’Argenio and J.-P. Katoen, “A theory of stochastic systems. Part II: Process algebra”, In. *Information and Computation* 203, Elsevier Inc, 2005, pp. 39–74.

[18] P. R. D’Argenio and J.-P. Katoen, “A theory of stochastic systems. Part I: Stochastic automata“, In. *Information and Computation* 203, Elsevier Inc, 2005, pp. 1–38.

[19] P. W. Glynn, “A GSMP Formalism for Discrete Event Systems”, In. *Proceedings of the IEEE*, Vol. 77, No. 1, January 1989, pp. 14-23.

[20] P. Šůcha, M. Kutil, M. Sojka, and Z. Hanzálek, “TORSCHE Scheduling Toolbox for Matlab”. In. *IEEE International Symposium on Computer-Aided Control Systems Design* (CACSD'06), Munich, Germany. 2006.

[21] S. Balsamo and A. Marin, “Queueing Networks”, In. M. Bernardo and J. Hillston (Eds.): *Formal Methods for Performance Evaluation* 2007, Lecture Notes in Computer Science 4486, Springer-Verlag Berlin Heidelberg, 2007, pp. 34–82.

[22] W. J. Stewart, "Performance Modelling and Markov Chains", In. M. Bernardo and J. Hillston (Eds.): *Formal Methods for Performance Evaluation*, LNCS 4486, Springer-Verlag, 2007, pp. 1–33.

Mokdad AROUS graduated in Computer Science at the University of Mentouri of Constantine, Algeria, and he obtained his BEng degree in June 2003, and his MSc degree in June 2007. Currently, he is Assist Professor at the University of Souk-Ahras (Algeria), and he prepares his PhD thesis in computer science at the MISC Laboratory (Laboratoire de Modélisation et Implémentation des Systèmes Complexes) of University of Mentouri of Constantine, Algeria. His research interests are in the area of formal specification, verification and performance evaluation of stochastic concurrent systems

Jean-Michel Ilié obtained several degrees in electronics and informatics among with its PhD thesis from the UPMC University of Paris (1990). The fields of his research concern the formal verification and validation of complex embedded systems. He received a special award as a co-author of the best paper for QEST 05 and contributes to four international books dedicated to its research domains.

Djamel Eddine Saidouni was born in Algeria in 1968. In 1996, he received his PHD degree in theoretical computer science from the university Paul Sabatier of Toulouse, France. Actually he is a professor at the department of computer science of Mentouri University of Constantine, Algeria. Also, he is the head of the CFSC research group of MISC laboratory. His main research domain interests formal models for specifying and verifying critical systems, real time systems, true concurrency models and state space explosion problem.