# Maximality Semantics based Stochastic Process Algebra for Performance Evaluation

Mokdad Arous
MISC Laboratory,
Mentouri University,
Constantine, 25000. Algeria
mokdad.a@gmail.com

Djamel-Eddine Saidouni
MISC Laboratory,
Mentouri University,
Constantine, 25000. Algeria
saidounid@hotmail.com

Jean-Michel Ilié
Lip6 laboratory,
Paris6 University,
Paris, France
jean-michel.ilie@lip6.fr

*Abstract*— **We introduce a true concurrency stochastic process algebra called S-LOTOS, such that stochastic durations of actions relate to general distributions. A structural operational semantics is defined so that the underlying model called MSLTS, extends the Maximality-based Labeled Transition Systems [1]. S-LOTOS is also viewed as a high level representation to specify Generalized Semi-Markov Processes (GSMP) in order to evaluate the system performances.**

***Keywords— Performances Evaluation, Stochastic Process Algebras, Generalized Semi-Markov Processes, Maximality-based Semantics.***

## I. 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 relate to random timing. In transmission systems for instance, transmission errors and decision changes in the traffic flow products, randomly lead to various communication delays. We need adequate stochastic timing models for specification and verification of these systems characterized by stochastic behaviors.

Several models were already developed to capture randomly varying time instants and also time intervals, among with Queuing models, e.g. [2], and stochastic versions of Petri Nets, e.g. [3, 4], Automata (Network), e.g. [5, 6], and Process Algebras, e.g. [7, 8].

In this paper, we refer to Stochastic Process Algebras (SPAs) that were proposed as a language for performance and dependability modeling and evaluation. Basically with each possible action, a random variable is introduced to represent its duration. Random variables are mostly assumed to be exponentially distributed, therefore the underlying semantics model turns out to be a Continuous Time Markov Chain (CTMC). Hence, performance measures can be extracted from the model automatically, by the use of efficient analytic techniques, e.g. [9, 21].

The restriction to exponential distribution allows defining an integrated semantics for SPAs through the interleaving-based approach [7, 11, 12]. The counter part is a loss of information about the concurrency execution of actions [13]. Moreover, for general distributions, the interleaving approach does not hold [10]. In fact, a more convenient model comes from a tune combination of true concurrency semantics used to reflect the parallel execution of actions, and the consideration of general distributions to specify the durations of actions, since more expressive than exponential distributions.

In the literature, several authors have proposed different approaches to incorporate general distributions in a process algebraic framework. Most famous are TIPP [14], IGSMP [15, 16], GSMPA [15, 17], ET-LOTOS [18] and SPADES (♠) [19, 20]. All these extensions of SPAs accords with ST-semantics (State-Transition based) to handle concurrent executions of actions with generally distributed durations. However, they consider actions as non atomic, enhancing the execution duration of any action by a clear separation of two events: the beginning and its end. Thus in semantic models, the progression of a delay is represented as a combination of two events: the delay starting and its termination. By identifying each delay with a unique name, causal dependencies are introduced such that the termination of a delay is uniquely related to one delay starting.

Alternatively, we propose to support general distributions in PA under the true concurrency maximality semantics [1]. Our aim is making possible to deal with true concurrency without being attacked by the state space explosion problem inherent to the splitting of actions. Form a modeling point of views, such an approach showed interest in easing the modeling stages.

The paper is organized as follows: Section 2 summarizes the idea of SPAs and their use for performance evaluation under exponential distributions. Section 3 sketches the reasons which make both general distributions and true concurrency paradigms preferable. Section 4 presents the Generalized Semi Markov Processes (GSMPs) which are suitable for representing event concurrency with generally distributed durations. In Section 5, we propose a new specification language, namely S-LOTOS, that derives from Basic LOTOS and its (true concurrency) maximality semantics. Syntax and semantics derivation rules are formally defined, then, to formalize the operational semantics of S-LOTOS expressions, we propose a specifically Labeled Transition System, namely Maximality-based Stochastic Labeled Transition System (MSLTS). From such semantic model, we show how a GSMP is obtained directly, by abstraction of the functional information. Section 6 concludes the paper and gives some further development.

## II. Stochastic Process Algebras for performance evaluation

SPAs are specific process algebras such that every action is assumed to be associated with a random variable representing its duration. An action is then described as a pair $(a, \lambda)$, where $a$ is the type of the action and $\lambda$ is the parameter of the distribution function governing its duration. This parameter determines the duration of the action probabilistically.

In case of exponential distribution (i.e. Markovian), the term "$(a, \lambda)$;" for some process $p$ is interpreted as "$p$ offers action $a$ within $t$ time units with probability $(1-e^{-\lambda t})$, then $p$ can evolve to deal with other actions". Such kinds of SPAs provide high-level specification formalism for CTMCs, and are called *Markovian Process Algebras* (MPAs). The semantic models of MPAs are transition systems wherein transitions are labeled with pairs $(a, \lambda)$ of actions and rates. By keeping only the rate information, a CTMC is obtained so that performance measures can be analytically extracted using traditional techniques. More details on the most widely used classes of Markovian models, solutions and applications, can be found in [2, 9, 21]. For SPAs, please refer to [7, 8, 10].

## III. Exponential distribution vs. General distributions for Performance Evaluation

Most standard performance evaluation formalisms, e.g. [4, 7, 12, 22], had focused on exponential distributions, mainly because the memoryless property for exponential distributions yields analytically tractable models, in particular Continuous-Time Markov Chains (CTMCs). In addition, under an interleaving approach, MPAs found easily the way to define their own operational semantics [7, 11, 12]. Anyway, let us point the limit by considering a distributed system composed of two processes s.t. $E$ executes $a$ in parallel with $b$, and $F$ executes either $a$ followed by $b$ or $b$ followed by $a$. Standardly, this is syntactically specified as follow:

$E= (a, \lambda)\ ;\ stop\ \ |[]|\ \ (b, \mu)\ ;\ stop$

$F = (a, \lambda)\ ;\ (b, \mu)\ ;\ stop\ []\ \ (b, \mu)\ ;\ (a, \lambda)\ ;\ stop$

Under the interleaving semantics, the Transition Systems which derive from these two expressions appear to be isomorphic (see Fig. 1), therefore both processes are considered as equivalent from the functional and also from the performance point of views [7, 11, 12]. In fact, due to the memoryless property of the exponential distribution, if we assume that $E$ completes $a$ before $b$, then the residual time to the completion of $b$ is still exponentially distributed with rate $\mu$, so the rate labeling the transition: ($stop$ |[]| $(b,\mu)$;$stop \rightarrow stop$ |[]| $stop$ ) is $\mu$ itself instead of $\mu$ conditional on $\lambda$ [7, 11, 12, 13]. The problem is here that standard interleaving approach does not yield any information about the fact that actions $a$ and $b$ are executed in parallel [13]. However, with respect to true concurrency paradigms, we will see further that the two systems are not equivalent.

Moreover, for general distributions the interleaving law does not hold [10]. In fact, consider now the process $E$ with assumption that durations of action $a$ and $b$ are generally distributed, the behavior representation of $E$ in Fig. 1 would be incorrect, since the time distribution for the transition: $stop$|[]|$(b, \mu)$; $stop \rightarrow stop$|[]|$stop$ should not be denoted by the parameter $\mu$. In the initial state action $a$ and $b$ are both enabled in parallel, but action $a$ happened to be executed first. Thus the time needed to complete $b$ in $stop$|[]|$(b,\mu)$;$stop$ should reflect the time already taken to complete action $\alpha$.

So the expressions $E$ and $F$ are not equivalent with regard to the memory property for delays in general distributions [19]. In such case of concurrent actions, it would be important to record the elapsed time in the intermediate states in order to know the residual time of the remaining actions.

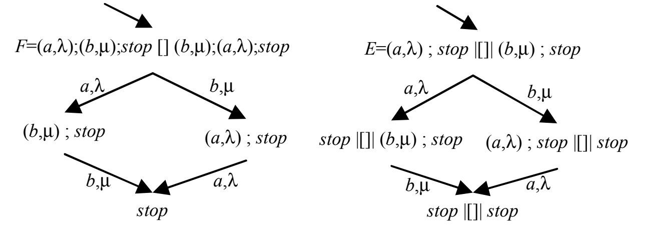

Figure 1. Behaviors of $E$ and $F$ according to the interleaving semantics

In other hand, exponential distributions are not realistic for modeling many phenomena in an adequate way [19], and the modeling paradigms suffer from a strong limitation in expressiveness. In fact, the exponential distribution is the most adequate only if the mean value of random variable is known. However, several events cannot be modeled using exponential distributions, for examples; if only the minimum and maximum of some quantity is known and more information is not available, the uniform distribution (on the interval between these bounds) would be a good choice, and deterministic delays like clock cycles in computers and timeouts in communication protocols are fixed, and they have deterministic distributions.

Since the exponential assumption is not always realistic, we need then a model with general distributions for specifying and analyzing stochastic systems. In fact, the focus on non-exponential distributions has been flourished and discussed, e.g. [15, 20]. However, if we allow transitions to be delayed by non-exponential distributions, the Markov property does not hold anymore, and the resulting stochastic process is then non-Markovian.

Up to our knowledge, the only candidates for representing systems with generally distributed event durations are Semi-Markov Processes (SMPs) [23] and above all the extension called Generalized Semi-Markov Processes (GSMP) [24]. In fact, SMPs are not enough expressive [17, 25]. Nowadays probabilistic approaches make use of GSMPs since capable to represent the semantics of parallel activities under generally distributed durations [16, 17].

Extending SPAs with generally distributed time is very important to make it really capable to modeling real concurrent systems. However, the interleaving semantics and the underlying expansion law are not adequate when considering action durations with general distributions. Combining therefore true concurrency maximality semantics, which does not obey expansion law and reflects true concurrency, with generally distributed timed actions, we can naturally model concurrent systems. Our goal is to support general distributions in SPA under the maximality semantics; this makes the formalism more expressive and more interesting from a practical viewpoint.

## IV. Generalized Semi-Markov Processes

GSMP was introduced as stochastic processes for modeling complex phenomena. GSMPs generalize CTMCs by allowing the action durations to be generally distributed and such that a timing not only depends on the current state but also on the past states. A GSMP is a probabilistic timed system, where durations of actions are expressed by means of random variables with a general probability distribution, and such that transitions are triggered by the occurrence of randomly timed events. A set of active events is associated to each state, these are the events that can start execution and cause the execution of transitions. The remaining time until the possible occurrence of an event is determined by clock; we have thus one clock per event. Clocks are initialized according to a continuous probability distribution function and run backwards all with the same pace. An event which is scheduled but does not initiate a transition is either abandoned or it is associated with the next state and its clock just continues running. GSMPs are formally defined as follows.

***Definition 1.*** Generalized Semi-Markov Process.

Let $DF$ be the set of (continuous) probability distribution functions ($\Re \rightarrow [0, 1]$). A GSMP is a tuple ($S$, $s_0$, $E$, $Activ$, $\mapsto$ , $K$, $F$) where:

- $S$ is a finite set of states, at least including the initial state $s_0$.
- $E$ is a (finite) set of events.
- $Activ$: $S \rightarrow 2^E$ is an injective function mapping each state with a subset of events, declared active in the state.
- $\mapsto \subseteq S{\times}E{\times}S$ is the set of transitions where $s \overset{e}{\mapsto} s'$ means that if event $e$ occurs first in $s$ then the process moves to state $s'$. Note that the process is deterministic, in other words, if $(s, e, s_1) \in \mapsto$ and $(s, e, s_2) \in \mapsto$ then $s_1 = s_2$.
- $K$: $S \rightarrow 2^E$ is the event setting function which represents all the events that are initialized when we reach a state, note that $K(s) \subseteq Activ(s)$ for all $s \in S$.
- $\forall s \in S$, the function $F_s : K(s) \rightarrow DF$ is an $s$-dependant function that assigns the event distribution functions (The distribution of an event depends on the state it was initialized in). □

## V. A Maximality Based Stochastic Process Algebra with General Distributions for Performance Evaluation

The objective of our SPA is the performance modeling of concurrent systems in true concurrency semantics, i.e. the maximality based semantics, instead of interleaving semantics which can not naturally model concurrent systems in the assumption of generally distributed action durations.

### A. Maximality semantics for basic LOTOS

The interleaving semantic is not a true concurrency semantic and it is inconsistent with the attribution of durations (with general distributions) to actions. In this section we present the principle of maximality semantics which allow to model accurately parallel execution of actions with durations of arbitrary distributions. We consider that the reader is familiar with the syntax of Basic LOTOS.

#### 1) The principle of maximality semantics

In the approach based on the maximality [1], transitions are events which represent only the beginning of action executions. Consequently, the concurrent execution of multiple actions is possible, and we can distinguish sequential and parallel executions of actions.

Given that several actions that have the same name can run in parallel, to distinguish the performances of each action an identifier is associated at the beginning of each execution of action, i.e. the transition or the associated event. In a state, an event is said maximal if it is the beginning of the execution of action that may possibly be still running in this state. Associate names of maximal events to states lead to the notion of configuration [26]. A configuration is denoted as $_M[F]$ where $M$ is the set of maximal events and $F$ is the behavior expression in the state. Transitions are labeled by $_M a_x$ where $a$ is an action, $x$ the name of the event identifying the beginning of execution of $a$, and $M$ is the set of maximal events corresponding to the direct causes of this transition. For example, the derivation graphs of behavior expressions $E$ and $F$ (introduced earlier) obtained by applying the maximality semantics are represented in Fig. 2, these graphs are called: *Maximality-based Labeled Transition System* (MLTS). The Structured operational semantics of maximality of Basic LOTOS and the notion of configuration are defined in [1, 26].

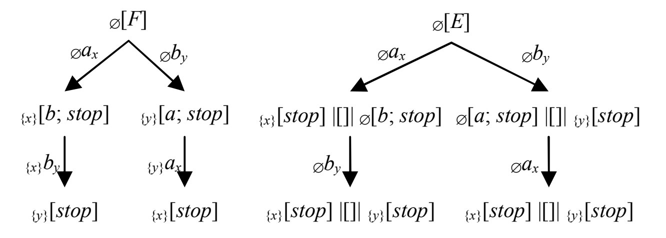

Figure 2. Behaviors of $E$ and $F$ according the maximality semantics

### B. S-LOTOS

For our approach, an action is represented by a pair ($a$, $f$), where $a$ is the action type and $f$ is the probability distribution function that governing the duration of action $a$. The basic idea is to use clock variables for maximal events to keep track of duration and to control and observe the passage of time. Since in our context the action durations are random, clocks are in fact random variables. Transitions are labeled with events which represent action starts, and for each event we associate a clock which appears in the resulting state with this maximal event. When a clock is set, it takes a random value whose probability depends on the distribution function of the action duration. As time evolves, clocks count down synchronously, i.e. all do so at the same rate. When a clock has expired, i.e.

have reached the value 0, next transitions are enabled. The labeled transition system obtained by extension with maximal events and stochastic clocks will be called *Maximality-based Stochastic Labeled Transition System* (MSLTS).

In the MSLTS example of Fig. 3, representing the behavior of expression $E = (a, f)$; *stop* |[]| $(b, g)$; *stop*, initially we have no maximal event. If the action $(a, f)$ happens first with the event name $x$ then a clock $X$ is initialized, according to the distribution function $f$, and start counting down. If in this stat the action $(b, g)$ happens with the event name $y$ then a state is reached in which clock $Y$ is initialized, according to the distribution function $g$, and start counting down while clock $X$ continue recording the time of action $a$. A symmetric scenario happens when action $(b, g)$ start first.

In the following, we formally present our language, called S-LOTOS (for Stochastic LOTOS), which we introduce in this article. S-LOTOS is an extension of basic LOTOS (based on the timed extension D-LOTOS [26]).

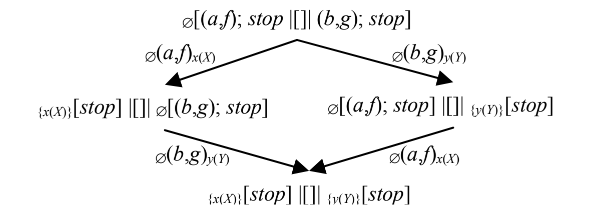
$_{\varnothing}[(a,f); stop$ |[]| $(b,g); stop]$
$_{\varnothing}(a,f)_{x(X)}$
$_{\varnothing}(b,g)_{y(Y)}$
$_{\{x(X)\}}[stop]$ |[]| $_{\varnothing}[(b,g); stop]$
$_{\varnothing}[(a,f); stop]$ |[]| $_{\{y(Y)\}}[stop]$
$_{\varnothing}(b,g)_{y(Y)}$
$_{\varnothing}(a,f)_{x(X)}$
$_{\{x(X)\}}[stop]$ |[]| $_{\{y(Y)\}}[stop]$

Figure 3. The SMLTS of: $E = (a,f)$; *stop* |[]| $(b,g)$; *stop*

*1) Syntax of terms and informal semantics of operators*

Let ς a countable set of clocks which is ranged over by $\{X, Y,\ldots\}$, and $DF$ the set of (continuous) probabilistic distribution functions ($\Re \rightarrow [0, 1]$) which is ranged over by $\{f, g,\ldots\}$.

Consider the behavior expression $(a, f)$; $(b, g)$; *stop*. In the initial state, no action is running and the corresponding configuration is then $_{\varnothing}[(a, f); (b, g); stop]$. From this state, the action $(a, f)$ can be started running with stochastic duration of distribution function $f$ whose event name is $x$. The transition $_{\varnothing}[(a, f); (b, g); stop] \xrightarrow{_{\varnothing}(a,f)_{x(X)}} {}_{\{x(X)\}}[(b, g); stop]$ is possible, where a clock variable $X$ is associated with the maximal event $x$ taking a random value according to the distribution function $f$ and starts counting. The resulting configuration shows that the action $b$ can begin its execution after the expiration of the $X$.

The syntax of S-LOTOS is defined as follows:

***Definition 2.*** Syntax of S-LOTOS.

Let $PN$ the set of process variables ranged over by $X$, and $A$ the set of observable actions ranged over by $a$, $b$,… A particular observable action $\delta \notin A$ is used to notify the successful termination of processes. $L$ denotes any subset of $A$. The internal action is denoted by $i$. The set of all actions is designated by $Act$ ($Act = A \cup (i, \delta)$). $\mathcal{B}$ ranged over by $E$, $F$... denotes the set of behavior expressions whose syntax is:

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

The set of event names is $\mathcal{M}$, ranged over by $x$, $y$,... $2_{fn}{}^{\mathcal{M}\times\varsigma}$ being the set of finite subsets of $\mathcal{M}\times\varsigma$. $MS$, $NS$... denote finite subsets of $\mathcal{M}\times\varsigma$. We define the notion of Stochastic Configurations as follows:

***Definition 3.*** Stochastic Configurations.

The set $SC$ of stochastic configurations is given by:

- $\forall\ E \in \mathcal{B}, \forall\ MS \in 2_{fn}{}^{\mathcal{M}\times\varsigma} : {}_{MS}[E] \in SC$
- $\forall\ P \in PN, \forall\ MS \in 2_{fn}{}^{\mathcal{M}\times\varsigma} : {}_{MS}[P] \in SC$
- if $\mathcal{E} \in SC$ then : $hide\ L\ in\ \mathcal{E} \in SC$
- if $\mathcal{E}, \mathcal{F} \in SC$ then : $\mathcal{E}\,[]\,\mathcal{F} \in SC$, $\mathcal{E}\,|[L]|\,\mathcal{F} \in SC$
- if $\mathcal{E} \in SC$ and $\{a_1, ..., a_n\}$, $\{b_1, ..., b_n\} \in 2_{fn}{}^{A}$ then: $\mathcal{E}[b_1/a_1, ..., b_n/a_n] \in SC$ □

*2) Structured operational semantics*

The transition relation between stochastic configurations is denoted $\rightarrow$. $\rightarrow\ \subseteq SC\times AtmS\times SC$, where the set of atoms of support $(Act\times DF)$ is $AtmS = 2_{fn}{}^{\mathcal{M}\times\varsigma}\times(Act\times DF)\times(\mathcal{M}\times\varsigma)$. For any subset of pairs of event name-clock $MS \in 2_{fn}{}^{\mathcal{M}\times\varsigma}$, $(a,f) \in (Act\times DF)$ and $(x, X) \in (\mathcal{M}\times\varsigma)$, the atom $(MS, (a, f), (x, X))$ will be denoted $_{MS}(a, f)_{x(X)}$. The choice of an event name and the corresponding clock can be done deterministically by using any function $get{:}2^{\mathcal{M}\times\varsigma}\text{-}\{\varnothing\} \rightarrow \mathcal{M}\times\varsigma$ satisfying $get(MS) \in MS$ for all $MS \in 2^{\mathcal{M}\times\varsigma}\text{-}\{\varnothing\}$. The operational semantics of S-LOTOS is given in Table 1. The definition of function ψ is given in the definition 4.

Consider configuration $_{MS}[stop]$, unlike the process *stop* in LOTOS, this configuration represents potential evolutions in function of maximal events in this configuration. The evolution stops as soon as all these actions terminate, i.e. all clocks expire, which is characterized by the predicate *Wait*: $2_{fn}{}^{\mathcal{M}\times\varsigma} \rightarrow \{true, false\}$ defined on any $MS \in 2_{fn}{}^{\mathcal{M}\times\varsigma}$ by: $Wait(MS) = \exists(x, X) \in MS$ such that $X > 0$. Intuitively $Wait(MS)=true$ if there is at least one running action referenced in $MS$. Once actions indexed by $MS$ have completed their execution, the successful termination may occur with deterministic duration zero.

In the configuration $_{MS}[(a, f); E]$, action $(a, f)$ start its execution after the expiration of all clocks concerning the current maximal events. Once action $a$ has become enabled, the transition to another state can take place where the resulting set of maximal events is $\{(x, X)\}$ meaning the initialization of the clock $X$ (by instantiation of the function $f$) which start counting. The same semantics for $_{MS}[(i, f); E]$ with internal action.

For parallel composition two situations are distinguished; if a process carries out an action $a \notin L$, it does so autonomously, the clocks of the other process does not affected by this transition, they continue counting. In case of a synchronization takes place, some action $a \in L$ is to start, both involved processes must be ready to perform $a$. For our language we consider the type of synchronization known as patient communication. That is, synchronization takes place as soon as all partners are ready to participate, i.e., once both clocks of events participating in synchronization have expired (The random variable that determines the duration to the synchronized action equals the maximum of the random variables in the individual processes).

Process $E[]F$ behave like either $E$ or $F$ (but not both), $E[b_1/a_1, \ldots , b_n/a_n]$ behave like $E$ except that actions $a_1, ..., a_n$ are renamed by $b_1, ..., b_n$ respectively. Process $hide\ L\ in\ \mathcal{E}$ behaves as $\mathcal{E}$ except that any activities of types within the set $L$ are *hidden*. Instead they appear as an internal action *i*.

TABLE I. STRUCTURED OPERATIONAL SEMANTICS OF S-LOTOS

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

$$\frac{\neg(wait(M))}{{}_{MS}[(a,f);E] \xrightarrow{{}_{MS}(a,f)_{x(X)}} {}_{\{(x,X)\}}[E]} \qquad (x,X) = get(\mathcal{M}\times\varsigma)$$

$$\frac{\neg(wait(M))}{{}_{MS}[(i,f);E] \xrightarrow{{}_{MS}(i,f)_{x(X)}} {}_{\{(x,X)\}}[E]} \qquad (x,X) = get(\mathcal{M}\times\varsigma)$$

$$\frac{\mathcal{E} \xrightarrow{{}_{MS}(a,f)_{x(X)}} \mathcal{E}' \quad a \notin L}{\mathcal{F}\,|[L]|\,\mathcal{E} \xrightarrow{{}_{MS}(a,f)_{x(X)}} \mathcal{F}\,|[L]|\,\mathcal{E}'}$$

$$\frac{\mathcal{E} \xrightarrow{{}_{MS}(a,f)_{x(X)}} \mathcal{E}' \quad a \notin L}{\mathcal{E}\,|[L]|\,\mathcal{F} \xrightarrow{{}_{MS}(a,f)_{x(X)}} \mathcal{E}'\,|[L]|\,\mathcal{F}}$$

$$\frac{\mathcal{E} \xrightarrow{{}_{MS}(a,f)_{x(X)}} \mathcal{E}' \quad \mathcal{F} \xrightarrow{{}_{NS}(a,g)_{y(Y)}} \mathcal{F}' \quad a \in L}{\mathcal{E}\,|[L]|\,\mathcal{F} \xrightarrow{{}_{MS\cup NS}(a,h)_{z(Z)}} \mathcal{E}'[z/x]\,|[L]|\,\mathcal{F}'[z/y]}$$
$$(z,Z) = get(\mathcal{M}\times\varsigma - ((\psi(\mathcal{E}) \cup (\psi(\mathcal{F})))$$

$$\frac{\mathcal{E} \xrightarrow{{}_{MS}(a,f)_{x(X)}} \mathcal{E}'}{\mathcal{F}[]\,\mathcal{E} \xrightarrow{{}_{MS}(a,f)_{x(X)}} \mathcal{E}'}$$

$$\frac{\mathcal{E} \xrightarrow{{}_{MS}(a,f)_{x(X)}} \mathcal{E}'}{\mathcal{E}[]\mathcal{F} \xrightarrow{{}_{MS}(a,f)_{x(X)}} \mathcal{E}'}$$

$$\frac{\mathcal{E} \xrightarrow{{}_{MS}(a,f)_{x(X)}} \mathcal{E}' \qquad a \notin L}{hide\,L\,in\,\mathcal{E} \xrightarrow{{}_{MS}(a,f)_{x(X)}} hide\,L\,in\,\mathcal{E}'}$$

$$\frac{\mathcal{E} \xrightarrow{{}_{MS}(a,f)_{x(X)}} \mathcal{E}' \qquad a \in L}{hide\,L\,in\,\mathcal{E} \xrightarrow{{}_{MS}(i,f)_{x(X)}} hide\,L\,in\,\mathcal{E}'}$$

$$\frac{\mathcal{E} \xrightarrow{{}_{MS}(a,f)_{x(X)}} \mathcal{E}' \quad a \notin \{a_1, \dots, a_n\}}{\mathcal{E}[b_1/a_1, \dots, b_n/a_n] \xrightarrow{{}_{MS}(a,f)_{x(X)}} \mathcal{E}'[b_1/a_1, \dots, b_n/a_n]}$$

$$\frac{\mathcal{E} \xrightarrow{{}_{MS}(a,f)_{x(X)}} \mathcal{E}' \qquad a = a_i \quad 1 \leq i \leq n}{\mathcal{E}[b_1/a_1, \dots, b_n/a_n] \xrightarrow{{}_{MS}(b_i,g)_{x(X)}} \mathcal{E}'[b_1/a_1, \dots, b_n/a_n]}$$

### C. Semantics models for S-LOTOS expressions

In this section, we define the operational semantics of S-LOTOS expressions by introducing a new labeled transition system, named Maximality-based Stochastic Labeled Transition System (MSLTS).

The structure of an MSLTS inherits from a previous true concurrency-based structure called Maximality-based Labeled Transition System (MLTS) [1].

In an MSLTS, a management of clocks is added to determine from each of the state, the actions that are running simultaneously accordingly to their durations.

An MSLTS is defined as follows:

***Definition 4.*** Maximality-based Stochastic Labeled Transition System (MSLTS).

Let $\mathcal{M}$ be a countable set of event names.

A Maximality-based Stochastic Labeled Transition System is a structure $(\Omega, A, DF, L, \mu, \xi, \psi)$ 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. $\alpha$ and $\beta$ are two functions: $T\rightarrow S$ mapping every transition with its source $\alpha(t)$ and its target $\beta(t)$.
- $A$ is a (finite) set of actions.
- $DF$ is a finite set of probability distribution functions ($\Re \rightarrow [0, 1]$).
- L: $T \rightarrow (A\times DF)$ : this function associates each transition with a pair composed of an action and a probability distribution function, thus $(\Omega, (A\times DF))$ is a transition system labeled by alphabet $(A\times DF)$.
- $\psi$: $S \rightarrow 2_{fn}^{\mathcal{M}\times\varsigma}$: this function associates every state with a finite set of pairs composed of a maximal event name and its corresponding clock in the state.
- $\mu$: $T\rightarrow 2_{fn}^{\mathcal{M}\times\varsigma}$: this function associates every transition with a finite set of pairs, composed of a maximal event name and its corresponding clock of actions that have started their execution so that their terminations, i.e. expiration of clocks, allow the start of this transition. This set corresponds to the direct causes of this transition.
- $\xi$: $T\rightarrow\mathcal{M}\times\varsigma$: this function associates each transition with an event name identifying an event occurrence and its corresponding clock, such that for any transition $t \in T$ :
  $\mu(t) \subseteq \psi(\alpha(t))$ ,
  $\xi(t) \notin \psi(\alpha(t)) - \mu(t)$ and
  $\psi(\beta(t)) = (\psi(\alpha(t)) - \mu(t)) \cup \{\xi(t)\}$ □

From an MSLTS, one can derive two semantics 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 duration of actions, whereas the performance model is obtained by abstracting the functional information.

### D. Building of the underlyied GSMP

We show now that the performance model of an MSLTS is a GSMP. The interest for S-LOTOS specification is that further analyses over the GSMP structures can yield performances evaluations, accordingly to the works [24, 25].

***Definition 5.*** Derivation of GSMP from an MSLTS.

Let $(\Omega, A, DF, \mu, \xi, \psi)$ be an MSLTS s.t. $\Omega = (S, s_0\ T, \alpha, \beta)$, then its underlying GSMP $(S', s'_0, E, F, Activ, \mapsto, K)$ is described as follows :

- $S' = S,\ s'_0 = s_0$ *and* $E = A$.
- For all $t\in T$ s.t. $L(t)=(a,f)$, $F_s(a)=f\in DF$ / $\alpha(t)=s$, $a\in Activ(\alpha(t))$ and $(\alpha(t), a, \beta(t))\in \mapsto$ .
- For all $s \in S$, for all $t \in T$ s.t. $s = \beta(t)$, $K(s)=Activ(s) - A'$ / $A' = \cup_{s'=\alpha(t)} (Activ(s') +$ $\{a / s=\alpha(t'), L(t')=a , \mu(t') \cap Activ(s') \neq \emptyset\})$. □

For example, The underlying GSMP for the S-LOTOS expression $E$ specified in Fig. 3 obtained by application of rules of definition 5 is represented in Fig. 4.

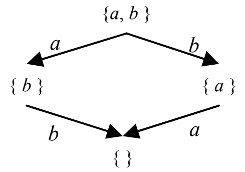

Figure 4. Underlying GSMP of expression $E = (a, f)$; *stop* |[]| $(b, g)$; *stop*

## VI. Conclusion and perspectives

In this paper we have defined a new Stochastic Process Algebra called S-LOTOS, to extend existing timed LOTOS specification language. Among the interests, arbitrarily distributed durations are now allowed, that means action durations governed by general distributions. Moreover, S-LOTOS conforms true concurrency paradigm as it was proposed for maximality-based semantic model.

As a formal work, we defined the operational semantics of S-LOTOS like a Maximality-based Stochastic Labeled Transition System (MSLTS). Such a semantic model introduces clocks to control the random time of the different events. From an MSLTS structure, we showed that GSMP structure can be derived, therefore S-LOTOS can be considered as a high-level specification formalism for GSMPs.

Our next study in the context of S-LOTOS will concern performance equivalence and bisimulation properties. Another point of interest would consist to augment the expressiveness of the language. We think to incorporate and study the impacts of other complex operators, like probabilistic alternative choice or interruption operators [1, 26].

## References

[1] 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.

[2] S. Balsamo and A. Marin, *Queueing Networks*, In. M. Bernardo and J. Hillston (Eds.): Formal Methods for Performance Evaluation 2007, LNCS 4486, Springer-Verlag Berlin Heidelberg, 2007, pp. 34–82.

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

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

[5] A. Benoit, B. Plateau, and W.J. Stewart, *Réseaux d'automates stochastiques à temps discret*, Revue des sciences et technologies de l'information, Technique et science informatiques "Evaluation de Performances", 24(2-3), 2005, pp. 229-248.

[6] 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.

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

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

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

[10] E. Brinksma and H. Hermanns, *Process algebra and Markov chains*, In. E. Brinksma, H. Hermanns, and J.-P. Katoen (Eds.): Lectures on Formal Methods and Performance Analysis, LNCS 2090, Springer-Verlag Berlin Heidelberg, 2001, pp. 183–231.

[11] M. Bernardo and R. Gorrieri, *Extended Markovian Process Algebra*, In. Ugo Montanari and Vladimiro Sassone (Eds.): CONCUR'96: Concurrency Theory (7th International Conference, Pisa, Italy, August 1996), Volume 1119 of LNCS . Springer.

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

[13] M. Bernardo, L. Donatiello and R. Gorrieri, *Integrating performance and functional analysis of concurrent systems with EMPA*, Technical report UBLCS-95-14, university of Bologna (Italy), September 1995 (revised March 1996).

[14] N. Götz, U. Herzog, and M. Rettelbach, Multiprocessor and distributed system design: The integration of functional specification and performance analysis using stochastic process algebras, In. L. Donatiello and R. Nelson, (Eds.): Performance Evaluation of Computer and Communication Systems, LNCS 729, Springer, 1993, pp. 121–146.

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

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

[17] 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 Vol. 1466, Proceedings of the 9th International Conference on Concurrency Theory (CONCUR'98), Nice (France), 1998, pp. 405-422.

[18] M. Ajmone Marsan, A. Bianco, L. Ciminiera, R. Sisto and A. Valenzano, *A LOTOS extension for the performance analysis of distributed systems*, IEEE/ACM Transactions on Networking, Vol. 2 No. 2, 1994, pp.151–164.

[19] 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 Berlin Heidelberg, 2001, pp. 375–429.

[20] 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.

[21] 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 Berlin Heidelberg, 2001, pp. 38–83.

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

[23] J. Janssen and R. Manca, *Applied Semi-Markov Processes*, Springer US, 2006.

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

[25] F. K. Dankar, *Approaches to Analysis and Simplification of non-Markovian System Models*, PhD. Thesis, School of Information Technology and Engineering (SITE), Faculty of Engineering, University of Ottawa, Canada, 2008.

[26] 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.