Real Time and Stochastic Time

Mario Bravetti

Università di Bologna, Dipartimento di Scienze dell'Informazione
Mura Anteo Zamboni 7, 40127 Bologna, Italy
E-mail: bravetti@cs.unibo.it

Abstract. We present a theory for the design and analysis of concurrent/distributed systems with real-time and stochastic time aspects. We start by presenting the model of Interactive Generalized Semi-Markov Processes (IGSMP): a compositional model for representing the class of stochastic processes known as Generalised Semi-Markov Processes (GSMPs), i.e. probabilistic timed systems where durations of delays are expressed by random variables with a general probability distribution. Technically, IGSMPs extend GSMPs with action transitions representing the ability of a process to interact with another process. Then, we introduce the calculus of Interactive Generalized Semi-Markov Processes, a stochastic process algebra which produces IGSMPs as semantic models of its terms. This is obtained by expressing the concurrent execution of delays through a simple probabilistic extension of Van Glabbeek and Vaandrageer's ST semantics based on dynamic names. We also present observational equivalence over IGSMPs, we observe that it is a congruence for all the operators of the calculus and we produce an axiomatization for this equivalence which is complete over finite-state strongly guarded processes. Finally, we present a case study on queuing systems $G/G/1/q$.

1 Introduction

The development of a software product requires, similarly as the other industrial products, an engineering cycle which is mainly composed of an initial specification stage and of a successive implementation stage. By using formal methods, it is possible to develop non-ambiguous specifications of systems which are analyzable with automated software tools. In particular, it is possible to verify the correctness of such specifications with respect to a certain set of requirements or to carry out a preliminary analysis of system performance, so to detect the design decisions that give the best results.

The importance of considering the behavior of concurrent systems with respect to time during their design process has been widely recognized [3, 28, 23, 21, 2, 4, 22, 18, 10, 5]. In particular two different approaches for expressing and analyzing time properties of systems have been developed which are based on formal description paradigms.

A first approach is devoted to the *evaluation of the performance* of concurrent systems (see e.g. [23, 21, 4, 22, 18, 10, 5]). According to this approach the time

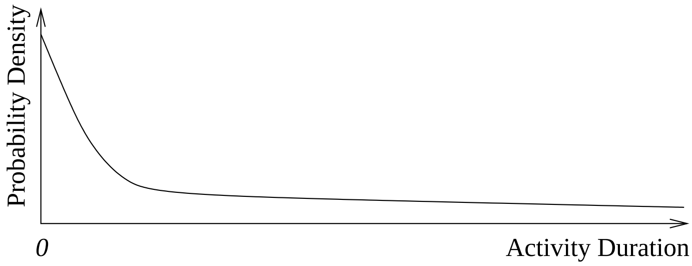


Fig. 1. Stochastic Time (Markovian) Activity

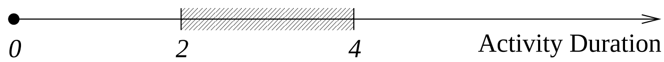


Fig. 2. Real-Time Activity

spent by a system in a certain activity is expressed probabilistically through a distribution of duration. Performance measures of systems can then be evaluated via mathematical or simulative techniques. This approach has led to the definition of *stochastic process algebras*, an extension of standard process algebras where a distribution of duration is associated with each action of a process. In most cases, as in [23, 21, 4, 22, 10], the expressiveness of such algebras is limited to exponential distributions of time, because this causes the passage of time to be "memoryless". As a consequence it is possible to completely avoid explicitly representing durations in semantic models. Moreover the limitation to *exponential distributions* allows for a straightforward transformation of the semantic model of a system into a *Continuous Time Markov Chain* (CTMC), a stochastic process which is easily mathematically analyzable for deriving performance measures. For this reason they are called *Markovian process algebras*. It is worth noting that the limitation imposed over durations is very strong because not even deterministic (fixed) durations can be expressed.

A second approach concentrates on the aspect of *real-time*, i.e. the expression of time constraints and the verification of exact time properties (see [3, 28] and the references therein). By this approach the parts of the system that are critical from the viewpoint of time bounds can be validated during the design phase through techniques such as e.g. *model checking* [3]. In this view *timed automata* have been developed by extending standard labeled transition systems with the representation of time by means of *clocks*. The time value assumed by a clock in a timed automata increases as time passes. In timed automata we have transitions representing the setting of a clock with a certain time value and transitions which can be executed provided that clocks satisfy a certain time constraint (see e.g. [3, 28]).

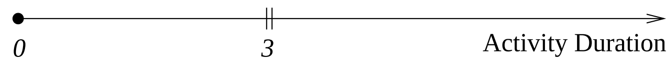


Fig. 3. Activity with a Fixed Duration

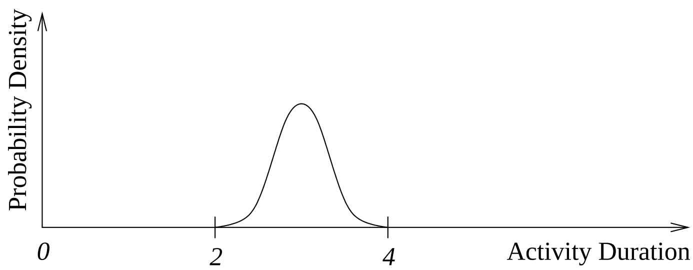


Fig. 4. Generally Distributed Activity

Using stochastic models with generally distributed probabilistic time provides a way for expressing and analyzing both stochastic time and real-time aspects of systems (see [6, 5]). The different aspects of time expressed by the Stochastic Time and Real-Time approaches can be seen as being *orthogonal*.

According to the first approach the possible values for the duration of an activity are quantified through probabilistic (exponential) distributions, but no time constraint is expressible: all duration values are possible with probability greater than zero. In Fig. 1 we depict the probability density for the duration values of an activity with an exponentially distributed duration.

According to the second approach some interval of time is definable for doing something, but the actual time the system spends in-between interval bounds is expressed non-deterministically. For instance, in Fig. 2 we depict an activity whose duration must be between 2 and 4 time units. Note that activities with a deterministic (fixed) duration are expressed when interval bounds coincide. For instance, in Fig. 3 we depict an activity whose duration is certainly 3.

A specification paradigm capable of expressing both aspects of time should be able of expressing both time constraints and a probabilistic quantification for the possible durations which satisfy such constraints. We obtain such an expressive power by considering stochastic models capable of expressing *general probability distributions* for the duration of activities. In this way time constraints are expressible via probability distribution functions that associate probability greater than zero only to time values that are possible according to the constraints. Technically, the set of possible time values for the duration of an activity is given by the *support* of the associated duration distribution. This idea of deriving real-time constraints from distribution supports was introduced in [6]. For instance, in Fig. 4 we depict an activity with a distribution whose support is the interval of Fig. 2. Note that with this approach we can also represent deterministic durations via trivial distribution functions that give all the probability to a single value of time.

Representing the real-time and probabilistic-time in a single specification paradigm allows us to model a concurrent system more precisely by expressing and analyzing the relationships between the two aspects of time. Moreover, the capability of expressing general distributions gives the possibility of producing much more realistic specifications of systems. System activities which have an uncertain duration could be represented probabilistically by more adequate distributions than exponential ones (e.g. Gaussian distributions or experimentally determined distributions).

This paper surveys the contents of [9, 14] and [5] chapters 6 and 7. In particular, we present (in Sect. 2) the model of Interactive Generalized Semi-Markov Processes (IGSMP) [14, 9, 5], a compositional model for representing real-time and stochastic time in concurrent/distributed systems. Conceptually, IGSMPs are an extension of Generalised Semi-Markov Processes (GSMPs), i.e. probabilistic timed systems where durations of delays are expressed by random variables with a general probability distribution, with action transitions representing the ability of a process to interact with another process. The technical machinery for IGSMPs can be seen as a probabilistic extension of that used in the pure real-time context for Timed Automata. In the case an IGSMP specification is complete from both the interaction and performance viewpoints then it is possible to formally derive the underlying GSMP for stochastic analysis purposes. Then, we present (in Sect. 3) the calculus of Interactive Generalized Semi-Markov Processes [14, 5], a stochastic process algebra which produces IGSMPs as semantic models of its terms. Technically, such an algebra is designed by expressing the concurrent execution of delays through a simple probabilistic extension of the dynamic name technique we introduced in [13] for expressing Van Glabbeek and Vaandrageer's ST semantics. We also show that observational equivalence over IGSMPs is a congruence for all the operators of the calculus and we produce an axiomatization for this equivalence which is complete over finite-state strongly guarded processes. Moreover, we present (in Sect. 4) a case study on queuing systems G/G/1/q, i.e. queuing systems with one server and a FIFO queue with q-1 seats, where interarrival time and service time are generally distributed. We also show how to derive the performance model of such queuing systems: a Generalised Semi-Markov Process. Finally, we report (in Sect. 5) some concluding remarks. Proof of theorems can be found in [5].

2 Interactive Generalized Semi-Markov Processes

In this section we introduce the compositional model of Interactive Generalized Semi-Markov Processes (IGSMPs). First of all, let us briefly recall the definition of a Generalized Semi-Markov Processes (GSMPs) [26].

A GSMP is a stochastic process representable by a transition system of the following kind. There is a fixed set of *elements* (an analogue of the clocks in a timed automaton) each with an associated probabilistic duration expressed by a time probability distribution. Each state of the transition system has an associated set of *active* elements taken from the set above. Such elements "decay" at a certain rate (for the sake of simplicity in this paper we will assume that all elements decay at the same rate) and the system sojourns in the state until an active element *dies* (completes its duration). When this happens a probabilistic state change occurs via a transition labeled with the terminated element and the other elements continue their life in the reached state, thus carrying out their residual duration. Whenever a state is entered where a previously dead element re-becomes active then the element *is born* again and executed for another time period determined from its associated probability duration distribution.

The following constraints must be satisfied from a GSMP:

– When the process moves from a state to another, no more than one element can be born or die contemporaneously.
– The active elements that do not die in a state must be again active in every reachable successive state (they keep their residual duration).

Example 1. As we will further comment, Fig. 5 and Fig. 16 represent the following two GSMPs: Fig. 5 a GSMP which executes two elements δ_1 and δ_2 in parallel; Fig. 16 a GSMP which, first executes element i, then it executes, with probability "0.4" element δ_1, with probability "0.6" element δ_2. In the figures states are labeled with the associated set of active elements. Moreover in the bottom of figures we associate a Probability Distribution Function to each element. Note that while the transition labeled by i in Fig. 16 leads to a probabilistic state change, the other transitions of Fig. 5 and Fig. 16 lead to a successive state with probability 1 (hence the probability information and the little bar are omitted). ■

In this section, we start by defining IGSMPs as an extension of GSMPs with action transitions representing the ability of a process to interact with another process in a concurrent/distributed system. From an IGSMP which is complete both from the interaction and performance viewpoints it is possible to derive a GSMP by means of the formal procedure introduced in [5] and sketched in this section.

Then, we observe that, in order to make it possible to define a bisimulation based equivalence over IGSMPs as a simple probabilistic extension of standard weak bisimulation [27], it is essential to adopt a canonical form for names of clocks (elements) used in an IGSMP. As a consequence, we define well-named IGSMPs as IGSMPs which follow a fixed rule for the names of clocks which is based on the dynamic name approach we introduced in [13] in the context of classical ST semantics.

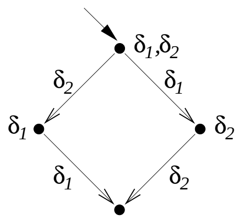
Moreover, we consider the problem of adequately developing a semantics for IGSMPs in terms of (continuously infinite state) probabilistic transition systems with numeric transitions representing time delays. In particular, we introduce Interactive Stochastic Timed Transition Systems (ISTTSs) to be used as semantic models for IGSMPs. As we will see, since IGSMPs introduce the capability of expressing non-deterministic choices in Generalised Semi-Markov Processes (GSMPs), it is important that the definition of such a semantics corresponds to a correct way of executing IGSMPs in the presence of adversaries resolving non-determinism. More precisely, we define the IGSMP semantics in such a way that the probabilistic duration of a clock is not decided all at once when the clock starts as done in [18], but step by step in each system state (in the theory of GSMPs this corresponds to recording spent lifetimes instead of residual lifetimes of delays). In this way an adversary cannot take decisions a priori, based on the knowledge he may get about the future behavior of the system.

This section is structured as follows. In Sect. 2.1 we present the basic concepts on which our approach is based. Then, in Sect. 2.2 we formalise the model of IGSMPs and the model of well-named IGSMPs. Moreover we show that the class of well-named IGSMPs is closed with respect to CSP parallel composition and hiding and we introduce a notion of weak bisimulation over well-named IGSMPs. Then, in Sect. 2.3 we sketch the formal procedure of [5] for deriving a performance model in the form of a *GSMP* from a complete IGSMP system specification. Moreover, in Sect. 2.4 we introduce the model of Interactive Stochastic Timed Transition Systems (ISTTSs), i.e. probabilistic transition systems (where probability is expressed by means of probability spaces over continuously infinite states) with numeric transitions representing time delays. We show that the class of ISTTSs is closed with respect to CSP parallel composition and hiding and we introduce a notion of weak bisimulation over ISTTSs. Finally, in Sect. 2.5 we present the semantics for IGSMPs which maps IGSMPs onto ISTTSs by recording spent lifetimes of delays. We show that weakly bisimilar IGSMPs give rise to weakly bisimilar semantic models and that the semantic mapping is compositional with respect to both CSP parallel composition and hiding.

2.1 Basic Concepts

The Basic Idea Some previous efforts have been made in order to develop models for general distributions [21, 2, 29]. With respect to such approaches, which somehow face this problem by starting from process algebra semantical viewpoint, in [11] we have introduced the idea that a specification paradigm expressing systems with generally distributed delays should originate from probabilistic models which are well-founded from the viewpoint of probability theory.

The stochastic processes mainly studied in the literature for performance evaluation purposes are in increasing order of expressivity: continuous time Markov chains (CTMCs), semi-Markov processes (SMPs), and generalized semi-Markov processes (GSMPs). The difference among them lies in the set of instants of process life which satisfy the Markov property, i.e. those instants such that the future behavior of the stochastic process depends only on the current state of the process and not on its past behavior. For CTMCs the Markov property holds in every instant of process life, for SMPs it holds only in the instants of state change, and for GSMPs it never holds, but can be retrieved through a different representation of process states (each state is turned into a continuous infinity of states) by the standard technique of [17] of introducing information about



$$PDF(\delta_1) = Det(5), \qquad PDF(\delta_2) = Gauss(0.5, 1)$$

Fig. 5. Parallel Execution of Generally Distributed Delays

spent or residual lifetimes. Since CTMCs can represent only activities with an exponentially distributed duration (only this distribution has the required memoryless property), the only candidates for representing systems with generally distributed durations are SMPs and GSMPs, and we now show that GSMPs are actually needed for our purposes. Consider the example of two delays δ_1 and δ_2 executed in parallel, the former with a deterministic duration 5 and the latter with a Gaussian duration with mean 0.5 and variance 1. This situation can be represented as in Fig. 5. In Fig. 5 each state is labeled with the set of delays which are in execution during the period of time the system sojourns in the state. In the beginning both delays are in contemporaneous execution and the system sojourns in the first state until one delay terminates. When this happens the system performs the transition labeled with the terminated action. Suppose that δ_1 terminates before δ_2 and the system reaches the state labeled with δ_2. In this state the delay δ_2 continues its execution until it terminates. As a consequence the sojourn time of the system in the state labeled with δ_2 (which is given by the residual distribution of delay δ_2) is not determined simply by the fact that the system is in this state, but depends on the time δ_2 has already spent in execution in the first state. In particular since we can no longer rely on the memoryless property (which is enjoyed only by exponential distributions) the residual distribution of δ_2 is different for every value of time spent. Therefore the process is not Markovian even in the instant when this state is entered.

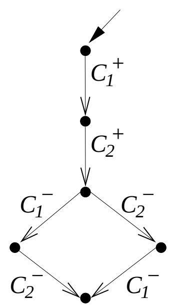
This example shows that even the simple case of two parallel delays with generally distributed durations cannot be represented by an SMP. The process of the example is, instead, a GSMP (δ_1 and δ_2 are its elements and state labels in Fig. 5 denote active elements). This can be seen as follows. If we imagine to give a different representation of the process where we replace the state labeled with δ_2 with infinitely many states each denoting a different spent lifetime for the delay δ_2, we can retrieve the Markov property. The sojourn time in each of the newly derived states would then be determined by the state itself (it would be given by the distribution of delay δ_2 conditioned on a particular value for the spent lifetime) and not by the previous behavior of the process. Another way to retrieve the Markov property would be to consider residual lifetimes instead of spent lifetimes.

Despite of the fact that in GSMPs the Markov property in instants of state change can be retrieved via an infinite state representation, when we consider the initial finite state system description (as that of Fig. 5) we have that the system behavior in a certain state indeed depends on what happened previously. In particular, delays executed in a state cannot be simply be considered as starting in that state as for CTMCs (or SMPs), but the may continue their execution from previous states. In other words a delay can no longer be considered as being executed *atomically* in a single transition, but we have to represent delays that start in a certain state, evolve through several states, and terminate in another state (in the previous example both δ_1 and δ_2 start in the first state and may terminate in another state). As a consequence the execution of a generally distributed delay must be characterized in models by the *two events* of *delay start* and *delay termination*.

In [21, 29] the Markovian representation of system behavior via delays "atomically" executed in transitions (which are labeled by the duration distribution of such delays) is adapted to systems with general distributions by adding to such transitions some information uniquely determining (through e.g. a pointer) the state where the delay labeling the transition is actually started. On the contrary, we prefer to stick to the notation of GSMPs, where representing the execution of generally distributed delays is simply done by: uniquely *identifying* each delay with a different *element* (similar to a clock of a timed automata), associating to each *element* the duration distribution of the delay it represents as in Fig. 5, assuming that element execution continues when going from state to state, and making explicit events of element start (or clock reset in timed automata). In this way we have a system representation which can be easily transformed into a GSMP (see Sect. 2.3) or a timed automata (see [5]) for analysis purposes.

In particular we represent temporal delays by clocks and we explicitly represent both basic events of start and termination of a clock C explicitly by means of a clock start transition C^+ and a clock termination transition C^-. On the other hand we do not explicitly indicate the set of clocks which are being executed in a state (the active elements in states of the GSMP represented in Fig. 5 as state labels) since such information can be easily inferred from its outgoing clock termination transitions. The resulting event-based representation of the system of Fig. 5 is depicted in Fig. 6, where delays δ_1 and δ_2 are represented by clocks C_1 and C_2, respectively, and we initially start both clocks by means of two explicit clock start transitions executed in zero time. To be precise, in our model we will consider clock start transitions labeled with pairs $\langle C^+, w\rangle$, where w is a *weight* used for representing probabilistic choices. In this way we reach the same expressivity as GSMPs.

A Model for Timed Concurrent Systems with General Distributions By following the "interactive" idea of [22], an IGSMP represents the behavior of a component by employing both *standard action transitions*, representing the interactive behavior of the component, and *clock start transitions* and *clock termination transitions*, representing the timed probabilistic behavior of the com-



$PDF(C_1) = Det(5), \qquad PDF(C_2) = Gauss(0.5, 1)$

Fig. 6. Event-Based Representation of Parallel Execution of Delays

ponent. Action transitions are just standard CCS/CSP transitions: when several action transitions are enabled in an IGSMP state, the choice among them is just performed non-deterministically and when IGSMPs are composed in parallel they synchronize following the CSP [24] approach, where the actions belonging to a given set S are required to synchronize. Clock start transitions are labeled with a clock name and a weight and represent the event of starting of a temporal delay whose probabilistic duration is given by the distribution associated with the clock. When several clock start transitions are enabled in an IGSMP state, the choice between them is performed probabilistically according to the weights of the transitions. Clock termination transitions are labeled with a clock name and represent the event of termination of the corresponding temporal delay. A system stays in a state enabling several termination transitions until one of the temporal delays currently in execution terminates and the corresponding transition is performed.

Besides IGSMPs, we also define *well-named IGSMPs*, a canonical form for names of clocks used in an IGSMP which is based on the dynamic name approach presented in [13] in the context of ST semantics. In particular, similarly as in [13] where the name associated with the two events of action start and action termination generated by the execution of an action is a pair consisting of the type a of the action and an index $i \in \mathbb{N}$, in a well-named IGSMP names n of clocks C_n consist of pairs "f, i", where f is the probability distribution associated with the clock (f is the "type" of the clock) and i an index. For each different distribution f, the index to be used when a new clock with distribution f starts is determined by a fixed rule: we choose the minimum index not currently used by the other clocks with the same distribution f already in execution [13]. The use of a fixed rule makes it possible to define equivalence as a simple probabilistic extension of standard weak bisimulation without having to associate clock names in the definition of equivalence.

We will define CSP parallel composition and hiding of well-named IGSMPs. As we will see, in order to obtain well-named IGSMPs as the result of such operations we have to suitably rename clocks. This is done by following the technique introduced in the dynamic name approach of [13].

IGSMPs may include both internal and external non-deterministic choices. While external non-deterministic choices may be resolved through synchronization with other system components (they are not present in system specifications which are complete from an interaction viewpoint), internal non-determinism represents an inherent underspecification of the system performance. Therefore adversaries (or schedulers) play an important role in the performance analysis of IGSMPs in that they allow internal non-determinism to be removed from an IGSMP, thus turning it, essentially, into a GSMP. In [5] a formal procedure is presented which turns IGSMPs which are complete both from the interaction and performance viewpoints into GSMPs. As we will see, this is done essentially by turning: *IGSMP* clocks into elements of the *GSMP* and *IGSMP* weighted choices into probabilistic choices of a *GSMP*.

Dealing with Non-determinism in Systems with General Distributions As we already explained, introducing non-determinism in probabilistic systems with general distributions causes new problems to arise with respect to the classical theory of GSMPs. Such problems derive from the interplay of non-deterministic choices and the probabilistic behavior of clocks when IGSMPs are actually executed. In particular, if we follow the classical approach of discrete event simulation (see e.g. [16]), in the instant a clock starts, the clock is set to a temporal value sampled from its duration distribution. As time passes the clock counts down and it terminates when it reaches value zero. From a technical viewpoint this means that, while the GSMP proceeds from state to state, we keep track of the quantity of time that clocks must still spend in execution (the *residual lifetimes* of the clocks). This approach to the execution of an IGSMP, which has been previously applied in [18] to systems including non-determinism and generally distributed time, has the drawback that an adversary can base its decisions (concerning non-deterministic choices) on the knowledge obtained a priori about the future behavior of the system, e.g. the information about the quantity of time that a delay will spend in execution.

In this paper we will consider an alternative approach to the execution of systems including non-determinism and generally distributed time which adequately handles non-deterministic choices. The idea is that we want the probabilistic duration of a generally distributed delay not to be decided all at once when the delay starts, but step by step in each system state. More precisely, this is realized by keeping track of the quantity of time spent by clocks in execution (*spent lifetimes* of clocks), and by evaluating, when a new IGSMP state is entered, the distribution of the residual duration of the clock from (i) the duration distribution associated with the clock, and (ii) the time it has already spent in execution. Such an approach, which is based on recording spent lifetimes instead of residual lifetimes, is adherent to the classical behavior of Timed Automata [28] where clocks are increased (and not decreased) while time passes. Besides it indeed solves the problem of executing a system with non-deterministic choices because, since the residual duration of clocks is sampled in every state traversed by the IGSMP, the adversary cannot gain a priori knowledge on the system behavior. Finally, considering spent lifetimes instead of residual lifetimes is correct also from a probabilistic viewpoint, because in probability theory the two approaches are both valid alternative ways to interpret a GSMP [17]. It is worth noting that the choice of adopting this alternative approach for representing the execution of an IGSMP is conceptual and not at all related with the technical differences between the formalism considered in [18] and IGSMPs. We could apply the technique used in [18] to IGSMPs as well.

Similarly as in [18], based on our approach to the execution of an IGSMP, we produce a semantics for IGSMPs which maps an IGSMP onto a transition system where: (i) the passage of time is explicitly represented by transitions labeled with numeric time delays and (ii) duration probability distributions are turned into infinitely branching probabilistic choices which lead to states performing numeric time delays with different durations. Differently from [18], we express semantic models of IGSMPs by means of "interactive" probabilistic timed transition systems which can be themselves composed and for which we define a notion of weak bisimulation. This allows us to develop a semantic mapping which is compositional with respect to parallel composition and hiding.

2.2 Definition of Interactive Generalized Semi-Markov Process

In this section we will present the model of Interactive Generalized Semi-Markov Processes (IGSMPs) and of well-named interactive generalized semi-Markov processes: a canonical form for IGSMPs which introduces some constraints on clock names and makes it simple to establish equivalence over IGSMPs.

The IGSMP Model The model of Interactive Generalized Semi-Markov Processes extends that of Generalized Semi-Markov Processes by expressing in addition to GSMP clocks (or elements) execution, also the execution of standard actions which can synchronize and have a zero duration. As far as probabilistic delays are concerned, they are modeled as in GSMPs by means of clocks C (which are like elements) whose duration is expressed through general probability distributions. In the following we will distinguish different clocks used in an IGSMP through "names", where C_n denotes the clock with name n. In an IGSMP the execution of a clock C_n is represented by means of two events: the event of clock start C_n^+ followed by the relative event of clock termination C_n^-. Therefore in an IGSMP we have three types of transitions: standard action transitions representing action execution, clock start transitions representing events C_n^+ and clock termination transitions representing events C_n^-. When a transition C_n^+ is performed by the IGSMP the clock C_n starts and continues its execution in every state traversed by the IGSMP. Whenever the clock C_n terminates, then the IGSMP executes the corresponding termination transition C_n^-. In particular, since, as in GSMPs, each started clock C_n which has not terminated yet must continue its execution in each state traversed by the IGSMP, all such states must have an outgoing transition C_n^-. Obviously clocks which can be simultaneously under execution in an IGSMP state must have different names (even if they have the same duration distribution), so that the event of termination of a clock C_n^- is always uniquely related to the corresponding event of start of the same clock C_n^+. Similarly as GSMPs, IGSMPs can also express probabilistic choices. This is obtained by associating with each start transition C_n^+ a weight $w \in \mathbb{R}^+$. In this way when a state of the IGSMP enables several clock start transitions $\langle C_n^+, w \rangle$, the choice of the clock to be started is performed probabilistically according to the weights w of the transitions. For instance, a state enabling two transitions labeled with $\langle C_n^+, w \rangle$ and $\langle C_{n'}^+, w' \rangle$ respectively starts clock C_n with probability $w/(w+w')$ and starts clock $C_{n'}$ with probability $w'/(w+w')$. On the other hand, IGSMPs also have, in addition to GSMPs, the capability of expressing non-deterministic choices. This because, as in standard labeled transition systems deriving from CCS/CSP terms, in the states of an IGSMP action transitions are just non-deterministically chosen. Alternative transitions labeled with invisible τ actions represent internal non-deterministic choices, while alternative transitions labeled with visible actions a (which are seen as incomplete potential transitions which wait for a synchronization with other system components) represent external non-deterministic choices which depend on the environment. An IGSMP represents a complete system (at least from the interaction viewpoint) only when it does not include any transition labeled by a visible action. Visible actions are synchronized via a CSP synchronization policy and a hiding operator is used to turn (synchronized) visible actions into complete invisible actions. This approach differs from that of the stochastic automaton model of [18], where two different kinds of semantics have to be defined in order to describe the *actual behavior* of closed systems and the *potential behavior* of open systems. In our approach both the potential and the actual behavior of the system are represented within the same model and complete systems are obtained by hiding all the actions of the model.

More precisely, in an IGSMP we have four different kinds of state:

- *silent states*, enabling invisible action transitions τ and (possibly) visible action transitions a only. In such states the IGSMP just performs a non-deterministic choice among the τ transitions in zero time and may potentially interact with the environment through one of the visible actions (see e.g. Fig. 7.a).
- *probabilistic states*, enabling $\langle C_n^+, w \rangle$ transitions and (possibly) visible action transitions a only. In such states the IGSMP just performs a probabilistic choice among the clock start transitions in zero time and may potentially interact with the environment through one of the visible actions (see e.g. Fig. 7.b).
- *timed states*, enabling C_n^- transitions and (possibly) visible action transitions a only. In such states the IGSMP executes all the clocks labeling the outgoing termination transitions according to their residual duration distribution. The clock that terminates first determines the transition to be performed. Note that since, as in GSMPs, we assume that clocks cannot terminate at the same

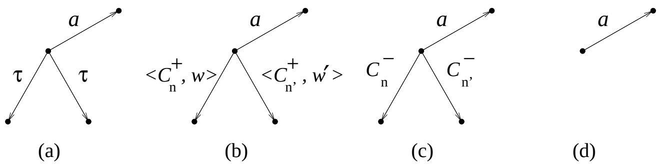


Fig. 7. Some examples of possible states of an IGSMP

instant, we always have a unique clock terminating before the other ones (see e.g. Fig. 7.c). While the IGSMP sojourns in the state, it may (at any time) potentially interact with the environment through one of the outgoing visible action transitions.
- *waiting states*, enabling standard visible actions only or no transition at all. In such states the IGSMP sojourns indefinitely. It may, at any time, potentially interact with the environment through one of the outgoing visible action transitions (see e.g. Fig. 7.d).

In the following we present the formal definition of Interactive Generalized Semi-Markovian Transition System (IGSMTS), then we will define Interactive Generalized Semi-Markov Processes as IGSMTSs possessing an initial state. Formally, we denote with PDF the set of probability distribution functions over $\mathbb{R}$ ranged over by $f, g, \ldots$ and with PDF^+ the set of probability distribution functions over $\mathbb{R}$ such that $f(x) = 0$ for $x < 0$ (representing duration distributions). Weights, belonging to $\mathbb{R}^+$, are ranged over by $w, w', \ldots$. Moreover, we denote the set of standard action types used in a IGSMTS by Act, ranged over by $\alpha, \alpha', \ldots$. As usual Act includes the special type τ denoting internal actions. The set $Act - \{\tau\}$ is ranged over by $a, b, \ldots$. The set of clocks of an IGSMTS is denoted by $\mathcal{C} = \{C_n \mid n \in \mathcal{C}Names\}$, where $\mathcal{C}Names$ is a set of clock names. Given a set $\mathcal{C}$, we denote with $\mathcal{C}^+ = \{\langle C_n^+, w\rangle \mid C_n \in \mathcal{C}, w \in \mathbb{R}^+\}$ the set of events denoting the starting of a clock and $\mathcal{C}^- = \{C_n^- \mid C_n \in \mathcal{C}\}$ the set of events denoting the termination of a clock. Let $\mathcal{C}^+ \cup \mathcal{C}^-$ be ranged over by $\theta, \theta', \ldots$. The set of states of an IGSMTS is denoted by Σ, ranged over by $s, s', \ldots$. We assume the following abbreviations that will make the definition of IGSMTSs easier. Let us suppose that $T \subseteq (\Sigma \times Labels \times \Sigma)$ is a transition relation, where $Labels$ is a set of transition labels, ranged over by l. We use $s \xrightarrow{l} s'$ to stand for $(s, l, s') \in T$, $s \xrightarrow{l}$ to stand for $\exists s' : s \xrightarrow{l} s'$, and $s \overset{l}{\not\longrightarrow}$ to stand for $\not\exists s' : s \xrightarrow{l} s'$.

Definition 1. An Interactive Generalized Semi-Markovian Transition System (IGSMTS) is a tuple $\mathcal{G} = (\Sigma, \mathcal{C}, D, Act, T_+, T_-, T_a)$ with

- Σ a set of states,
- $\mathcal{C}$ a set of clocks,
- $D : \mathcal{C} \longrightarrow PDF^+$ a function that assigns a duration probability distribution function to each clock,

- Act a set of standard actions,
- $T_+ \subseteq (\Sigma \times \mathcal{C}^+ \times \Sigma)$, $T_- \subseteq (\Sigma \times \mathcal{C}^- \times \Sigma)$, and $T_a \subseteq (\Sigma \times Act \times \Sigma)$ three transition relations representing clock start and termination events and action execution, respectively, such that: [1]

  1 $\forall s \in \Sigma.$
  $s \xrightarrow{\tau} \implies \not\exists \theta.\, s \xrightarrow{\theta}$

  2 $\forall s \in \Sigma.$
  $\exists C_n, w.\, s \xrightarrow{\langle C_n^+, w\rangle} \implies \not\exists C_{n'}.\, s \xrightarrow{C_{n'}^-}$

  3 $\exists \mathcal{S} : \Sigma \longrightarrow \mathcal{P}(\mathcal{C})$ the *active clock* function, such that $\forall s \in \Sigma.$

  a) - $s \xrightarrow{\alpha} s' \implies \mathcal{S}(s') = \mathcal{S}(s)$
  - $s \xrightarrow{\langle C_n^+, w\rangle} s' \implies \mathcal{S}(s') = \mathcal{S}(s) \cup \{C_n\}$
  - $s \xrightarrow{C_n^-} s' \implies C_n \in \mathcal{S}(s) \;\wedge\; \mathcal{S}(s') = \mathcal{S}(s) - \{C_n\}$

  b) $\exists C_n, w.\, s \xrightarrow{\langle C_n^+, w\rangle} \implies C_n \notin \mathcal{S}(s)$

  c) $C_n \in \mathcal{S}(s) \wedge s \not\xrightarrow{\tau} \wedge \not\exists C_{n'}, w.\, s \xrightarrow{\langle C_{n'}^+, w\rangle} \implies s \xrightarrow{C_n^-}$

  4 $\forall s \in \Sigma.$
  $s \xrightarrow{\langle C_n^+, w\rangle} s' \implies act(s') \subseteq act(s)$

  where the *enabled action* function $act : \Sigma \longrightarrow \mathcal{P}(Act)$ is defined by $act(s) = \{\alpha \mid s \xrightarrow{\alpha}\}$. ■

Definition 2. An Interactive Generalized Semi-Markov Process (IGSMP) is a tuple $\mathcal{G} = (\Sigma, \mathcal{C}, D, Act, T_+, T_-, T_a, s_0)$, where $s_0 \in \Sigma$ is the initial state of the IGSMP and $(\Sigma, \mathcal{C}, D, Act, T_+, T_-, T_a)$ is an IGSMTS such that function $\mathcal{S}$ in item 3 of Definition 1 also satisfies $\mathcal{S}(s_0) = \emptyset$. ■

The constraints over transition relations T_+, T_- and T_a guarantee that each state of the IGSMP belongs to one of the four kind of states above. In particular, the first requirement says that if a state can perform internal τ actions then it cannot perform events of clock starts or clock terminations. Such a property derives from the assumption of *maximal progress*: the possibility of performing internal actions prevents the execution of delays. The second requirement says that if a state can perform clock start events then it cannot perform clock termination events. Such a property derives from the assumption of *urgency of delays*: clock start events cannot be delayed but must be performed immediately, hence they prevent the execution of clock termination transitions. The third requirement checks that clock starting and termination transitions are consistent with the set of clocks that should be in execution in each state of the IGSMP. This is done by defining a function $\mathcal{S}$ which maps each state onto the expected set of

[1] For the sake of readability here and in the rest of the paper we assume the following operator precedence when writing constraints for transition relations: existential quantifier > "and" operator > implication.

clocks in execution, i.e. the set of clocks which have started but not terminated yet. In particular, in the initial state s_0 such a set is empty. The constraint a) defines the construction rule of the active clock set for each state reachable from s_0. In the case of a transition from a state s to a state s' labeled with a standard action, the active clocks of s' stem from the active clocks of s, as no clock can be terminated given that a standard action has been performed. If a transition from s to s' is labeled with a clock start event $\langle C_n^+, w\rangle$, then s' inherits the active clock set of s and adds to this set the started clock C_n. Finally, in the case of a transition from s to s' labeled with a clock termination event C_n^-, s' inherits the active clock set of s without such a terminated clock C_n. Constraints b) and c) concern the legality of the outgoing transitions of a state. In particular, the former says that the name of a clock labeling a starting transition must be fresh (i.e. no clock with such a name must be currently in execution). The latter says that each state without τ and $\langle C_n^+, w\rangle$ outgoing transitions must have a $C_{n'}^-$ outgoing transition for each active clock $C_{n'}$. This definition preserves both the maximal progress and the urgency of delays assumptions and, in each state where it is possible, guarantees the possibility of terminating each delay that is still active. The fourth requirement of Definition 1 implements the following constraint over the structure of IGSMPs which makes their theory simpler. The unique role of clock start transitions in an IGSMP must be to lead to a timed state where the started clocks are actually executed, hence the execution of such transitions cannot cause new behaviors to be performable by the IGSMP. As we will see in Sect. 3, such a constraint is satisfied by the semantic models of terms of the calculus of IGSMPs, hence we consider this constraint not to be really restrictive for our purposes. Formally, we require that the set of action transitions enabled after a clock start transition is a subset of (or equal to) the set of action transitions enabled before such a transition. This guarantees that no new behaviors can be introduced by clock start transitions because: (i) no new behavior beginning with a τ transition can be executable after a clock start transition (states enabling clock start transitions cannot enable τ transitions), and (ii) every potential behavior beginning with a transition a executable after a clock start transition can never be actually executed by hiding a, because before the clock start transition there is a potential behavior beginning with the same action a, which, when hidden, preempts the clock start (see the following subsection about the hiding of IGSMPs).

The Well-Named IGSMP Model As already explained, the model of well-named IGSMPs introduces a canonical form for names of clocks used in an IGSMP which is based on the dynamic name approach of [13] and makes it simple to develop an equivalence notion over IGSMPs which matches clocks with the same duration distribution.

The constraint on the use of clock names in an IGSMP that we consider concerns the names n which are used for clocks when they start. As we already explained the name used for a starting clock must be fresh, i.e. no clock with such a name must be currently in execution. The requirement that we now add is that the new clock name which is used must depend from the duration distribution f associated with the starting clock and from the names of the clocks (with the same distribution f) already in execution, according to a fixed rule. In particular, we take the set of clock names to be defined by $\mathcal{C}Names = (PDF^+ \times \mathbb{N}^+)$, where "$f, i$" is a name for a clock with associated distribution f. The name "f, i" which is used for a starting clock must be such that i is the least $i \in \mathbb{N}^+$ which is not used in the name of any clock with the same distribution f already in execution. Note that, using just duration distributions as clock names is not sufficient because indexes $i \in \mathbb{N}^+$ are needed in order to have different clock names when clocks with the same duration distribution are simultaneously executed.

By applying the dynamic name technique introduced in [13] to clock names, we have that, since the method to compute the index for a starting clock is fixed, clocks of systems that perform equivalent execution traces get the same names. As a consequence, when establishing equivalence of well-named IGSMPs we do not have to associate clock names but we can rely on a simple extension of standard bisimulation.

Since in a well-named IGSMP names for clocks cannot be chosen arbitrarily and the clock names which are considered make it clear by themselves which is the duration distribution associated with a clock, with respect to IGSMTSs (Definition 1), in the definition of well-named IGSMTSs we omit set $\mathcal{C}$ and function D.

Definition 3. A well-named Interactive Generalized Semi-Markovian Transition System is a tuple $\mathcal{G} = (\Sigma, Act, T_+, T_-, T_a)$ where Σ and Act are defined as in Definition 1, while the definition of the transition relations T_+, T_- and T_a is obtained from that given in Definition 1 by substituting the constraint $b)$ of item 3 with the stronger constraint:

$$b) \quad \exists C_{f,i}, w.\, s \xrightarrow{\langle C^+_{f,i}, w\rangle} \implies i = \min\{j \mid j \in \mathbb{N}^+,\, C_{f,j} \notin \mathcal{S}(s)\} \qquad \blacksquare$$

Note that the new version of constraint $b)$ guarantees that the name used for a starting clock is always fresh as required by the old version of constraint $b)$ (see Definition 1).

Definition 4. A well-named Interactive Generalized Semi-Markov Process is a tuple $\mathcal{G} = (\Sigma, Act, T_+, T_-, T_a, s_0)$, where $s_0 \in \Sigma$ is the initial state of the well-named IGSMP and $(\Sigma, Act, T_+, T_-, T_a)$ is a well-named IGSMTS such that function $\mathcal{S}$ in item 3 of Definition 1 also satisfies $\mathcal{S}(s_0) = \emptyset$. $\blacksquare$

As an important remark, we would like to point out that, since the rule expressed by constraint $b)$ of Definition 4 reuses the indexes i of terminated clocks, each IGSMP with a finite set of states can be transformed into a well-named IGSMP with a finite set of states, by renaming clocks.

Parallel of Well-Named IGSMPs Now, we address the problem of defining parallel composition à la CSP [24] of well-named IGSMPs, where the standard actions of a given set S are required to synchronize and the synchronization of two actions of type a is again an action of type a.

Intuitively, it should be clear that when composing in parallel two IGSMPs, a suitable renaming of the clocks is necessary in order to obtain an IGSMP, i.e. to preserve the requirements on transition relations of Definition 2. Indeed composing in parallel two IGSMPs could lead to some conflict concerning the identification of the clocks of the composed model through names. More precisely, we have to cope with a name conflict whenever two clocks with the same name "f, i" are simultaneously in execution in both IGSMPs. In such a case the same name identifies two different clocks by compromising the relationship between the start and termination events of the two clocks. When considering well-named IGSMPs instead of just IGSMPs we have in addition the problem of preserving the rule for the name of starting clocks expressed by constraint b) of Definition 4.

The solution that we adopt, which is taken from the dynamic name technique of [13], consists in using l and r (left and right) as references to the two well-named IGSMPs $\mathcal{G}', \mathcal{G}''$ which are composed in parallel by $\mathcal{G}' \parallel_S \mathcal{G}''$ and relating each clock name locally used in $\mathcal{G}'$ (or $\mathcal{G}''$) to the corresponding well-named IGSMP $\mathcal{G}'$ (or $\mathcal{G}''$) through the reference l (or r). In this way C_{f,l_i} (C_{f,r_i}) denotes the clock $C_{f,i}$ executed by $\mathcal{G}'$ ($\mathcal{G}''$). In order to obtain a well-named IGSMP, when building the composed model, such "extended" names are renamed so that the rule for the name of starting clocks expressed by constraint b) of Definition 4 is satisfied. For instance, let us suppose that both $\mathcal{G}'$ and $\mathcal{G}''$ execute a clock with the same duration distribution f. For both well-named IGSMPs in isolation we represent such an event by activating the clock $C_{f,1}$. Somehow in the composed model we have to distinguish such clocks through names because they can be simultaneously in execution. Let us suppose that in $\mathcal{G}' \parallel_S \mathcal{G}''$ the first delay with distribution f that starts is the one executed by $\mathcal{G}'$. According to the well-naming rule, in the composed model $\mathcal{G}' \parallel_S \mathcal{G}''$ such a clock must get name "$f, 1$". Hence we map $C_{f,1}$ to the "extended" name of the clock $C_{f,1}$ executed by $\mathcal{G}'$, thus creating the following mapping:

$$C_{f,1} \longrightarrow C_{f,l_1}$$

denoting that the first clock with distribution f of the composed model $C_{f,1}$ corresponds to the first clock with distribution f of the lefthand well-named IGSMP. Then, if the second clock to be executed is the clock $C_{f,1}$ belonging to the righthand well-named IGSMP, in the composed model we create the fresh name "$f, 2$" (according to the well-naming rule) and have in addition the following mapping:

$$C_{f,2} \longrightarrow C_{f,r_1}$$

In Table 1 we present an example of execution of a composed model $\mathcal{G}' \parallel_S \mathcal{G}''$ by showing how the mapping function (between the clock names of the composed model $\mathcal{G}' \parallel_S \mathcal{G}''$ and the corresponding clock names locally used in $\mathcal{G}'$ and $\mathcal{G}''$) for clocks with distribution f evolves.

By following such a procedure, we build the composed model by dynamically storing all current mappings between the clock names of the composed model

Well-named IGSMPs	*Composed Model*	*Mapping Function*
$\mathcal{G}'$ *starts* $C_{f,1}$	$\mathcal{G}' \parallel_S \mathcal{G}''$ *starts* $C_{f,1}$	$C_{f,1} \longrightarrow C_{f,l_1}$
$\mathcal{G}''$ *starts* $C_{f,1}$	$\mathcal{G}' \parallel_S \mathcal{G}''$ *starts* $C_{f,2}$	$C_{f,1} \longrightarrow C_{f,l_1}$ $C_{f,2} \longrightarrow C_{f,r_1}$
$\mathcal{G}''$ *starts* $C_{f,2}$	$\mathcal{G}' \parallel_S \mathcal{G}''$ *starts* $C_{f,3}$	$C_{f,1} \longrightarrow C_{f,l_1}$ $C_{f,2} \longrightarrow C_{f,r_1}$ $C_{f,3} \longrightarrow C_{f,r_2}$
$\mathcal{G}''$ *ends* $C_{f,1}$	$\mathcal{G}' \parallel_S \mathcal{G}''$ *ends* $C_{f,2}$	$C_{f,1} \longrightarrow C_{f,l_1}$ $C_{f,3} \longrightarrow C_{f,r_2}$
$\mathcal{G}'$ *starts* $C_{f,2}$	$\mathcal{G}' \parallel_S \mathcal{G}''$ *starts* $C_{f,2}$	$C_{f,1} \longrightarrow C_{f,l_1}$ $C_{f,2} \longrightarrow C_{f,l_2}$ $C_{f,3} \longrightarrow C_{f,r_2}$

Table 1. Renaming of the clocks in $\mathcal{G}' \parallel_S \mathcal{G}''$

and the local clock names of the two well-named IGSMPs by employing a table (mapping function) for each distribution f. In general, when a clock $C_{f,i}$ with distribution f is started by one of the two composed well-named IGSMPs, we do the following: (i) we choose the first index j for the distribution f which is unused in the composed model (by checking the table related to the duration probability distribution f), and we use the name "f, j" for the clock in the composed model; (ii) we add to the table related to distribution f the mapping $C_{f,j} \longrightarrow C_{f,l_i}$ if the clock is executed by the lefthand well-named IGSMP or $C_{f,j} \longrightarrow C_{f,r_i}$ if the clock is executed by the righthand well-named IGSMP. When a clock $C_{f,i}$ with distribution f is terminated by one of the two composed well-named IGSMPs, we do the following: (i) we establish the name "f, j" associated with the terminating clock in the composed model by checking the table related to distribution f (it must include $C_{f,j} \longrightarrow C_{f,l_i}$ if the clock is executed by the lefthand well-named IGSMP or $C_{f,j} \longrightarrow C_{f,r_i}$ if the clock is executed by the righthand well-named IGSMP); (ii) we remove from the table related to the duration probability distribution f the mapping for the name "f, j" of the composed model.

Now we formally define the parallel composition $\mathcal{G}_1 \parallel_S \mathcal{G}_2$ of two well-named IGSMPs $\mathcal{G}_1$ and $\mathcal{G}_2$, where the synchronization set S is a subset of $Act - \{\tau\}$.

We denote with $Loc = \{l, r\}$, ranged over by loc, the set of locations, where l stands for left and r for right. We denote an *index association*, whose elements are associations (j, loc_i), with *iassoc*, which ranges over the set *IAssoc* of partial bijections from $\mathbb{N}^+$ to $Loc \times \mathbb{N}^+$. Moreover, a mapping M is a relation from PDF^+ to $\mathbb{N}^+ \times (Loc \times \mathbb{N}^+)$ such that $\forall f \in PDF^+ . M_f \in IAssoc$ [2], i.e. M is a mapping including an index association for each different duration distribution. We denote the set of mappings M by $\mathcal{M}$. In the following we use the shorthand $f : (j, loc_i)$ to stand for $(f, (j, loc_i)) \in M$ that represents the clock mapping $C_{f,j} \longrightarrow C_{f,loc_i}$. Finally we make use of the auxiliary function $n : IAssoc \longrightarrow \mathbb{N}^+$ that computes the new index to be used for a clock name according to the well-naming rule, by choosing the minimum index not used by the other clocks with the same

[2] Given a relation M from A to B, we denote with M_a the set $\{b \in B \mid (a, b) \in M\}$.

distribution already in execution, i.e. $n(iassoc) = min\{k \mid k \notin dom(iassoc)\}$.

Definition 5. The parallel composition $\mathcal{G}_1 \,\|_S\, \mathcal{G}_2$ of two well-named IGSMPs $\mathcal{G}_1 = (\Sigma_1, Act, T_{+,1}, T_{-,1}, T_{a,1}, s_{0,1})$ and $\mathcal{G}_2 = (\Sigma_2, Act, T_{+,2}, T_{-,2}, T_{a,2}, s_{0,2})$, with S being the synchronization set, is the tuple $(\Sigma, Act, T_+, T_-, T_a, (s_{0,1}, s_{0,2}, \emptyset))$ with:

- $\Sigma = \Sigma_1 \times \Sigma_2 \times \mathcal{M}$ the set of states,
- $T_+ \subseteq (\Sigma \times \mathcal{C}^+ \times \Sigma)$, $T_- \subseteq (\Sigma \times \mathcal{C}^- \times \Sigma)$, and $T_a \subseteq (\Sigma \times Act \times \Sigma)$ the least transition relations, such that $\forall (s_1, s_2, M) \in \Sigma$.

  $\mathbf{1_l}$ $s_1 \xrightarrow{\alpha} s_1',\ \alpha \notin S \implies (s_1, s_2, M) \xrightarrow{\alpha} (s_1', s_2, M)$

  $\mathbf{2}$ $s_1 \xrightarrow{a} s_1' \,\wedge\, s_2 \xrightarrow{a} s_2',\ a \in S \implies (s_1, s_2, M) \xrightarrow{a} (s_1', s_2', M)$

  $\mathbf{3_l}$ $s_1 \xrightarrow{\langle C^+_{f,i}, w\rangle} s_1' \,\wedge\, s_2 \stackrel{\tau}{\not\longrightarrow} \implies$
  $(s_1, s_2, M) \xrightarrow{\langle C^+_{f,n(M_f)}, w\rangle} (s_1', s_2, M \cup \{f : (n(M_f), l_i)\})$

  $\mathbf{4_l}$ $s_1 \xrightarrow{C^-_{f,i}} s_1' \wedge s_2 \stackrel{\tau}{\not\longrightarrow} \;\wedge\; \nexists C_{g,h}, w.\, s_2 \xrightarrow{\langle C^+_{g,h}, w\rangle} \wedge\, f:(j, l_i) \in M \implies$
  $(s_1, s_2, M) \xrightarrow{C^-_{f,j}} (s_1', s_2, M - \{f:(j, l_i)\})$

  and also the symmetric rules $\mathbf{1_r}, \mathbf{3_r}, \mathbf{4_r}$ referring to the local transitions of $\mathcal{G}_2$, which are obtained from the rules $\mathbf{1_l}, \mathbf{3_l}, \mathbf{4_l}$ by exchanging the roles of states s_1 (s_1') and s_2 (s_2') and by replacing l_i with r_i, hold true.
- $(s_{0,1}, s_{0,2}, \emptyset) \in \Sigma$ the initial state ■

Each state $s \in \Sigma$ of the composed model is represented by a triple including a pair of states ($s_1 \in \Sigma_1$ and $s_2 \in \Sigma_2$) and an auxiliary memory M containing all the index associations currently active in such a state. Rules **1** (**2**) describe the behavior of the composed model in the case of a standard action α performed by one (or both, via a synchronization) well-named IGSMPs, when $\alpha \notin S$ ($\alpha \in S$). Rules **3** and **4** define the behavior of the composed model in the case of delays locally performed by components. When in $\mathcal{G}_1$ ($\mathcal{G}_2$) occurs a transition labeled with $\langle C^+_{f,i}, w\rangle$, denoting the beginning of a delay with duration distribution f, then the new index $n(M_f)$ is determined for identifying the action at the level of the composed model, and the new mapping $f : (n(M_f), l_i)$ ($f : (n(M_f), r_i)$) is added to M. Conversely, when in $\mathcal{G}_1$ ($\mathcal{G}_2$) occurs a transition labeled with $C^-_{f,i}$, denoting the termination of a clock with duration distribution f, the particular clock with index j associated to l_i (r_i) in M_f terminates at the level of the composed model, and the index j becomes available. Note that the negative clauses in the premises enforce the maximal progress and the urgency of delays assumptions.

Example 2. Called $\mathcal{G}$ the well-named IGSMP of Fig. 8 which repeatedly executes delays with distribution f, we depict in Fig. 9 the well-named IGSMP $\mathcal{G} \,\|_\emptyset\, \mathcal{G}$. In Fig. 8 and 9 and in the following we consider C^+_n to be a shorthand for $<C^+_n, 1>$. ■

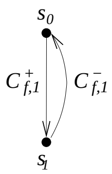


Fig. 8. Example of Well-Named IGSMP

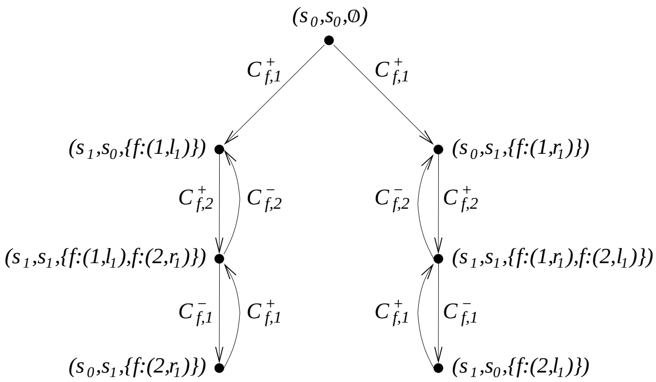


Fig. 9. Example of Parallel Composition of Well-Named IGSMPs

Theorem 6. *Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be two well-named IGSMPs. Then for each $S \subseteq Act - \{\tau\}$, $\mathcal{G}_1 \,\|_S\, \mathcal{G}_2$ is a well-named IGSMP.* ■

Hiding of Well-Named IGSMPs Now, we address the problem of defining hiding of well-named IGSMPs, where the standard actions of a given set L are turned into invisible τ actions.

As we already explained, the capability of hiding actions makes it possible to turn visible "incomplete" actions into invisible "complete" ones, thus giving the possibility of building a complete system from several system components. In particular while a visible action transition (as long as it is enabled) can delay indefinitely before being performed, when such an action is turned into an invisible action it must be executed in zero time.

Now we formally define the hiding $\mathcal{G}/L$ of a well-named IGSMP $\mathcal{G}$, where the set L of the visible actions to be hidden is a subset of $Act - \{\tau\}$.

Definition 7. The hiding $\mathcal{G}/L$ of a well-named IGSMP $\mathcal{G} = (\Sigma, Act, T_{+,1}, T_{-,1}, T_{a,1}, s_0)$ with L being the set of visible actions to be hidden is the tuple $(\Sigma, Act, T_+, T_-, T_a, s_0)$ where $T_+ \subseteq (\Sigma \times \mathcal{C}^+ \times \Sigma)$, $T_- \subseteq (\Sigma \times \mathcal{C}^- \times \Sigma)$, and $T_a \subseteq (\Sigma \times Act \times \Sigma)$ are the least set of transitions, such that $\forall s \in \Sigma$. [3]

[3] In order to distinguish transition of $T_{+,1}$, $T_{-,1}$ and $T_{a,1}$ from transitions of T_+, T_- and T_a we denote the former with "$\longrightarrow_1$" and the latter simply with "$\longrightarrow$".

1 $s \xrightarrow{\alpha}_1 s'$, $\alpha \notin L \implies s \xrightarrow{\alpha} s'$
2 $s \xrightarrow{a}_1 s'$, $a \in L \implies s \xrightarrow{\tau} s'$
3 $s \xrightarrow{\theta}_1 s' \wedge \not\exists a \in L.\, s \xrightarrow{a}_1 \implies s \xrightarrow{\theta} s'$ ■

Rules **1** and **2** are standard. Rule **3** says that the effect of the hiding operator over states of $\mathcal{G}$ which enable standard actions in L is to preempt all clock related transitions according to the maximal progress assumption.

Theorem 8. *Let $\mathcal{G}$ be a well-named IGSMP. Then for each $L \subseteq Act - \{\tau\}$, $\mathcal{G}/L$ is a well-named IGSMP.* ■

Equivalence of Well-Named IGSMPs Now we will introduce a notion of weak bisimulation over well-named IGSMPs. In particular such a notion matches start and termination events of clocks with the same duration distribution by using strong bisimulation, deals with probabilistic choices similarly as in [25], and abstracts from standard τ actions similarly as in [27]. In the following Sect. 3 we will see that weak bisimulation is a congruence with respect to both parallel composition and hiding operators.

In our context we express cumulative probabilities by aggregating weights.

Definition 9. Let $\mathcal{G} = (\Sigma, Act, T_+, T_-, T_a)$ be a well-named IGSMTS.The function $TW : \Sigma \times PDF^+ \times \mathcal{P}(\Sigma) \longrightarrow \mathbb{R}^+ \cup \{0\}$, which computes the aggregated weight that a state $s \in \Sigma$ reaches a set of states $I \in \mathcal{P}(\Sigma)$ by starting a delay with duration distribution $f \in PDF^+$ is defined as: [4]

$$TW(s, f, I) = \sum \{\!| \, w \mid \exists i \in \mathbb{N}^+, s' \in I.\ s \xrightarrow{\langle C^+_{f,i}, w\rangle} s' \, |\!\}$$

■

We are now in a position to define the notion of weak bisimilarity for well-named IGSMPs (in the style of that of [22]). Let $NPAct = Act \cup \mathcal{C}^-$, the set of non-probabilistic actions, be ranged over by σ. Besides, let $\stackrel{\sigma}{\Longrightarrow}$ denote $(\xrightarrow{\tau})^* \xrightarrow{\sigma} (\xrightarrow{\tau})^*$, i.e. a sequence of transitions including a single σ transition and any number of τ transitions. Moreover, we define $\stackrel{\hat{\sigma}}{\Longrightarrow} = \stackrel{\sigma}{\Longrightarrow}$ if $\sigma \neq \tau$ and $\stackrel{\hat{\tau}}{\Longrightarrow} = (\xrightarrow{\tau})^*$, i.e. a possibly empty sequence of τ transitions.

Definition 10. Let $\mathcal{G} = (\Sigma, Act, T_+, T_-, T_a)$ be a well-named IGSMTS. An equivalence relation β on Σ is a *weak bisimulation* iff $s_1 \,\beta\, s_2$ implies

– for every $\sigma \in NPAct$ and $s'_1 \in \Sigma$,
$s_1 \xrightarrow{\sigma} s'_1$ implies $s_2 \stackrel{\hat{\sigma}}{\Longrightarrow} s'_2$ for some s'_2 with $s'_1 \,\beta\, s'_2$,

[4] We use $\{\!|$ and $|\!\}$ to denote multiset parentheses. The summation of an empty multiset is assumed to yield 0. Since the method for computing the new index of a delay f that starts in a state P is fixed, we have that several transitions f^+ leaving P have all the same index i.

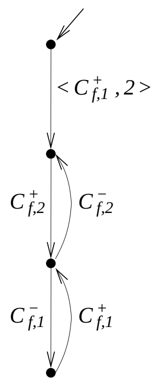


Fig. 10. Minimal Well-Named IGSMP

– $s_2 \stackrel{\hat{\tau}}{\Longrightarrow} s'_2$ for some s'_2 such that, for every $f \in PDF^+$ and equivalence class I of β,

$$TW(s_1, f, I) = TW(s'_2, f, I)$$

■

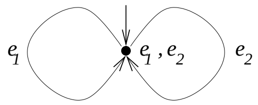
Example 3. In Fig. 10 we depict the minimal semantic model for the well-named IGSMP of Fig. 9, which is obtained by merging weakly bisimilar states. The weight 2 of the initial transition derives from the aggregation of the weights of the two initial transitions in the model of Fig. 18. However since in the initial state there is no alternative to such a transition, its weight is not relevant for the actual behavior (in isolation) of the IGSMP. ■

2.3 Deriving the Performance Model from an IGSMP

In this section we sketch the formal procedure of [5] for deriving a GSMP from an IGSMP system specification. In particular this transformation is possible only if the specification of the system is *complete* both from the *interaction* and from the *performance* points of view.

An IGSMP specification is complete from the interaction viewpoint if the system specified is not a part of a larger system which may influence its behavior, hence when every standard action appearing in the IGSMP is an *internal* τ action.

An IGSMP specification is complete from the performance viewpoint if all the choices in which the specified system may engage are quantified probabilistically. This means that the IGSMP must not include silent states actually leading to a non-deterministic choice among different future behaviors. In other words a silent state either must have only one outgoing τ transition, or all its outgoing τ transitions must lead to equivalent behaviors. This notion can be formally defined as follows: An IGSMP is complete w.r.t. performance if it can be reduced, by aggregating weakly bisimilar states (see Sect. 2.2), to a model without silent states.



$ElPDF(e_1) = f, ElPDF(e_2) = f$

Fig. 11. Derived GSMP

The formal procedure for the derivation of the GSMP is composed of three phases.

The first phase consists in *minimizing* the state space Σ of the Well-Named IGSMP $\mathcal{G}$ by aggregating states that are equivalent according to the notion of weak bisimulation defined in Sect. 2.2. Since we supposed that the $\mathcal{G}$ satisfies the two conditions above, a side effect of this minimization is that all τ actions disappear from $\mathcal{G}$.

The second phase is the transformation of every *probabilistic tree* present in the semantic model into a single probabilistic choice. First of all weights are turned into the corresponding probability values. A probabilistic tree is formed by the possible probabilistic paths that go from a given probabilistic state (the root of the tree) to a timed or waiting state (a leaf of the tree). Note that such trees cannot include loops composed of one or more transitions, because after each clock start the number of clocks in execution strictly increases. To be precise, such trees are directed acyclic graphs (*DAGs*) with root, since a node may have multiple incoming arcs. The probabilistic trees are flattened into a single choice that goes directly from the root to the leaves of the tree.

The final phase is the derivation of the GSMP. The set of elements of the *GSMP* is composed of the clocks $C_{f,i}$ labeling the transitions of $\mathcal{G}$. The states of the *GSMP* are the timed and waiting states of $\mathcal{G}$. A transition leaving a state of the *GSMP* is derived beginning from a clock termination transition leaving the corresponding timed state of $\mathcal{G}$ and, in the case this transition leads to a probabilistic state, a corresponding probabilistic state change occurs in the derived *GSMP*. Each transition of the *GSMP* is labeled by the element $C_{f,i}$ terminating in the corresponding termination transition.

Example 4. In Fig. 11 we show the *GSMP* derived, by applying the translation above, from the Well-Named IGSMP of Fig. 9. In particular the *GSMP* is obtained from the minimal model of Fig. 10, which is the result of the first phase. Since such model does not include standard action transitions the system considered is complete both from the interactive and the performance viewpoints. In the *GSMP* of Fig. 11 the states are labeled by the active elements and the transitions with the terminating elements. Each transition of the *GSMP* lead to a single successive state (the probabilistic state change after consists in trivially selecting that state with probability 1). The elements e_1 and e_2 represent the clocks $C_{f,1}$ and $C_{f,2}$ respectively, and the probability distribution function of both is given by function f. ■

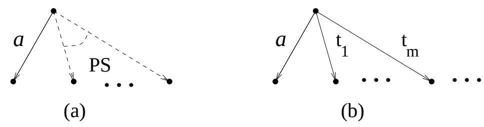


Fig. 12. Some examples of possible states of an ISTTS

2.4 Interactive Stochastic Timed Transition Systems

In this section we introduce Interactive Stochastic Timed Transition Systems (ISTTSs) that will be used in the following to define a semantics for IGSMPs.

The ISTTS Model In this section we formally define Interactive Stochastic Timed Transition Systems (ISTTS) which include three type of transitions: *standard action transitions*, representing the interactive behavior of a system component, *probabilistic transitions* (expressed by means of probability spaces [5]) representing (infinitely branching) probabilistic choices and *numeric time transitions* representing a fixed temporal delay.

As far as standard actions are concerned they have exactly the same behavior as in IGSMPs. In ISTTS non-deterministic choices can arise not only from transitions labeled with standard visible actions (like in IGSMPs), but also from transitions representing the passage of time. As usual in the real time literature (see e.g. [28]), several timed transition leaving a state offer the possibility to the observer to choose the amount of time after which he wants to observe the status of the system.

In ISTTS we have two different kinds of state:

- *silent states* which are exactly like in IGSMPs.
- *probabilistic states* enabling probabilistic transitions, expressed by a probability space PS, and (possibly) visible action transitions a only. In such states the ISTTS just chooses a new state in zero time according to the probability space and may potentially interact with the environment through one of its visible actions (see e.g. Fig. 12.a).
- *timed states* enabling numeric timed transitions t and (possibly) visible action transitions a only. In such states the ISTTS just performs a non-deterministic choice among the numeric timed transitions (which cause the amount of time labeling the transition to pass) and may potentially interact with the environment through one of its visible actions (see e.g. Fig. 12.b).

In the following we present the formal definition of Interactive Stochastic Timed Transition System (ISTTS), then we will define Rooted Interactive Stochastic Timed Transition Systems as ISTTSs possessing an initial state. Formally, given a time domain $TD \subseteq \mathbb{R}^+ \cup \{0\}$, we use $t, t', \ldots$, representing time values, to range over TD.

[5] See, e.g., [5] for a formal definition of probability spaces.

Definition 11. An Interactive Stochastic Timed Transition System (ISTTS) is a tuple $\mathcal{D} = (\Sigma, TD, Act, P, T_t, T_a)$ with

- Σ a possibly infinite set of states,
- TD a time domain, i.e. the set of possible values over which the labels of the numeric timed transitions range,
- Act a set of standard actions,
- $P : \Sigma' \to PS(\Sigma - \Sigma')$, where $\Sigma' \subset \Sigma$ and $PS(\Sigma'')$ denotes the family of probability spaces over sets of states $\Sigma''' \subseteq \Sigma''$, the probabilistic transition relation which associates a probability space with some of the states of the ISTTS; and $T_t \subseteq (\Sigma \times TD \times \Sigma)$ and $T_a \subseteq (\Sigma \times Act \times \Sigma)$ two transition relations representing time passage and action execution, respectively. P, T_t and T_a must be such that $\forall s \in \Sigma$.
  - $s \xrightarrow{\tau} \implies s \notin dom(P) \wedge \;\nexists t. s \xrightarrow{t}$
  - $s \in dom(P) \implies \nexists t. s \xrightarrow{t}$
  - $s \xrightarrow{\tau} \vee \exists t. s \xrightarrow{t} \vee s \in dom(P)$ ■

Definition 12. A Rooted Interactive Stochastic Timed Transition System (RISTTS) is a tuple $\mathcal{D} = (\Sigma, TD, Act, P, T_t, T_a, s_0)$, where $s_0 \in \Sigma$ is the initial state and $(\Sigma, TD, Act, P, T_t, T_a)$ is an ISTTS. ■

The meaning of the constraints over transition relations is the following. The first requirement says that (similarly as in IGSMPs) if a state can perform internal τ actions then it can perform neither probabilistic transitions nor timed transitions (*maximal progress* assumption). The second requirement says that (similarly as in IGSMPs) if a state can perform probabilistic transitions then it cannot perform timed transitions (*urgency of choices* assumption). The third requirement says that (similarly as in IGSMPs) we cannot have states where time is not allowed to pass (time deadlocks).

Parallel of Rooted ISTTSs Now we define, similarly as for IGSMPs, the parallel composition à la CSP of RISTTSs.

In such a parallel composition the numeric timed transitions of the composed RISTTSs are constrained to synchronize, so that the same amount of time passes for both systems, i.e. when time advances for one RISTTS it must also advance for the other RISTTS.

Definition 13. The parallel composition $\mathcal{D}_1 \,\|_S\, \mathcal{D}_2$ of two RISTTSs $\mathcal{D}_1 = (\Sigma_1, TD, Act, P_1, T_{t,1}, T_{a,1}, s_{0,1})$ and $\mathcal{D}_2 = (\Sigma_2, TD, Act, P_2, T_{t,2}, T_{a,2}, s_{0,2})$, with $S \subset Act - \{\tau\}$ being the synchronization set, is the tuple $(\Sigma, TD, Act, P, T_t, T_a, (s_{0,1}, s_{0,2}))$ with:

- $\Sigma = \Sigma_1 \times \Sigma_2$ the set of states

- P the partial function defined over $\Sigma_1 \times \Sigma_2$ obtained from P_1 and P_2 as follows: $\forall s_1 \in \Sigma_1, s_2 \in \Sigma_2$.
  $P(s_1, s_2) = Id^1_{s_2}(P_1(s_1))$ if $s_1 \in dom(P_1) \wedge s_2 \xrightarrow{t}\!\!\!\!\not\;$
  $P(s_1, s_2) = Id^2_{s_1}(P_2(s_2))$ if $s_2 \in dom(P_2) \wedge s_1 \xrightarrow{t}\!\!\!\!\not\;$
  $P(s_1, s_2) = P(s_1) \cdot P(s_2)$ if $s_1 \in dom(P_1) \wedge s_2 \in dom(P_2)$
  $P(s_1, s_2)$ is not defined otherwise
  with $Id^1_{s_2} : \Sigma_1 \longrightarrow (\Sigma_1 \times \{s_2\})$ defined by $\forall s \in \Sigma_1 . Id_{s_2}(s) = (s, s_2)$ and $Id^2_{s_1} : \Sigma_2 \longrightarrow (\{s_1\} \times \Sigma_2)$ defined by $\forall s \in \Sigma_2 . Id_{s_1}(s) = (s_1, s)$.
- $T_t \subseteq (\Sigma \times TD \times \Sigma)$ and $T_a \subseteq (\Sigma \times Act \times \Sigma)$ the least transition relations, such that
  $\mathbf{1_l}$ $s_1 \xrightarrow{\alpha} s'_1,\ \alpha \notin S \implies (s_1, s_2) \xrightarrow{\alpha} (s'_1, s_2)$
  $\mathbf{1_r}$ $s_2 \xrightarrow{\alpha} s'_2,\ \alpha \notin S \implies (s_1, s_2) \xrightarrow{\alpha} (s_1, s'_2)$
  $\mathbf{2}$ $s_1 \xrightarrow{a} s'_1 \wedge s_2 \xrightarrow{a} s'_2,\ a \in S \implies (s_1, s_2) \xrightarrow{a} (s'_1, s'_2)$
  $\mathbf{3}$ $s_1 \xrightarrow{t} s'_1 \wedge s_2 \xrightarrow{t} s'_2 \implies (s_1, s_2) \xrightarrow{t} (s'_1, s'_2)$
- $(s_{0,1}, s_{0,2}) \in \Sigma$ the initial state. ∎

The probability space associated by function P to the states of the composed model is determined as follows.

Whenever only a RISTTS, e.g. $\mathcal{D}_1$ in state s_1, engages in a probabilistic choice (and such a choice is not prevented by the maximal progress assumption) we evaluate the probability space in the composed model by considering the probability space $Id^1_{s_2}(P_1(s_1))$ "induced" by applying function $Id^1_{s_2}$ to the states in the domain of the probability space $P_1(s_1)$ (see, e.g., [5] for a formal definition of induced probability spaces). Whenever both RISTTSs engage in probabilistic choices, we produce a single global probability space by computing the "product" of the two probability spaces (see, e.g., [5] for a formal definition of product of probability spaces): a probability space whose domain is the cartesian product of the initial domains and where individual choices are assumed performed independently.

When evaluating action transitions we just make use of standard rules. Finally we require timed transitions to synchronize.

Theorem 14. *Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be two RISTTSs. Then for each $S \subseteq Act - \{\tau\}$, $\mathcal{D}_1 \,\|_S\, \mathcal{D}_2$ is a RISTTS.* ∎

Hiding of Rooted ISTTSs Now we define, similarly as for IGSMPs, the hiding of RISTTSs.

Definition 15. The hiding $\mathcal{D}/L$ of a RISTTS $\mathcal{D}_1 = (\Sigma, TD, Act, P_1, T_{t,1}, T_{a,1}, s_0)$, with $L \subset Act - \{\tau\}$ being the set of visible actions to be hidden, is the tuple $(\Sigma, TD, Act, P, T_t, T_a, s_0)$, with:

- P the partial function obtained from P_1 by removing from its domain those states (and the associated probability spaces) which enable at least one transition labeled with an action in L

– $T_t \subseteq (\Sigma \times TD \times \Sigma)$ and $T_a \subseteq (\Sigma \times Act \times \Sigma)$ the least transition relations, such that $\forall s \in \Sigma$. [6]

1 $s \xrightarrow{\alpha}_1 s', \ \alpha \notin L \implies s \xrightarrow{\alpha} s'$
2 $s \xrightarrow{a}_1 s', \ a \in L \implies s \xrightarrow{\tau} s'$
3 $s \xrightarrow{t}_1 \wedge \ \not\exists a \in L.\, s \xrightarrow{a}_1 \implies s \xrightarrow{t}$ ■

Similarly as for IGSMPs, in the definition of the hiding operator in addition to standard rules we make use of rules which enforce the maximal progress assumption.

Theorem 16. *Let $\mathcal{D}$ be a RISTTS. Then for each $L \subseteq Act - \{\tau\}$, $\mathcal{D}/L$ is a RISTTS.* ■

Equivalence of Rooted ISTTSs Now we introduce a notion of weak bisimulation for RISTTSs which constitutes an extension of the approach of [25] to probability spaces and abstracts from standard τ actions similarly as in [27].

Given an equivalence relation β on a set Σ and a set $I \subseteq \Sigma$, we first define the function $EC_{I,\beta} : I \to \Sigma/\beta$ which maps each state $s \in I$ into the corresponding equivalence class $[s]_\beta$ in Σ.

Definition 17. Let $\mathcal{D} = (\Sigma, TD, Act, P, T_t, T_a)$ be an ISTTS. An equivalence relation β on Σ is a *weak bisimulation* iff $s_1 \ \beta \ s_2$ implies

– for every $\alpha \in Act$,
$$s_1 \xrightarrow{\alpha} s_1' \text{ implies } s_2 \stackrel{\hat{\alpha}}{\Longrightarrow} s_2' \text{ for some } s_2' \text{ with } s_1' \ \beta \ s_2',$$
– for every $t \in TD$,
$$s_1 \xrightarrow{t} s_1' \text{ implies } s_2 \xrightarrow{t} s_2' \text{ for some } s_2' \text{ with } s_1' \ \beta \ s_2',$$
– $s_2 \stackrel{\hat{\tau}}{\Longrightarrow} s_2'$ for some s_2' with $EC_{\Sigma_{P(s_1)},\beta}(P(s_1)) \approx EC_{\Sigma_{P(s_2')},\beta}(P(s_2'))$

where Σ_{PS} denotes the set of states in the domain of the probability space PS and "$\approx$" denotes equivalence between probability spaces, i.e. coincidence of probabilities over the intersection of domains and zero probability for all states not included in such intersection (see [5] for a formal definition).

Two states s_1 and s_2 are weakly bisimilar, denoted by $s_1 \approx s_2$, iff (s_1, s_2) is included in some weak bisimulation. Two RISTTSs $(\mathcal{D}_1, s_{0,1})$ and $(\mathcal{D}_2, s_{0,2})$ are weakly bisimilar, if their initial states $s_{0,1}$ and $s_{0,2}$ are weakly bisimilar in the ISTTS obtained with the disjoint union of $\mathcal{D}_1$ and $\mathcal{D}_2$. ■

In the last item we exploit induced probability spaces to check that states s_1 and s_2' have the same aggregated probability to reach the same equivalence classes.

[6] In order to distinguish transition of $T_{t,1}$ and $T_{a,1}$ from transitions of T_t and T_a we denote the former with "$\longrightarrow_1$" and the latter simply with "$\longrightarrow$".

2.5 A Semantics for Interactive Generalized Semi-Markov Processes

In this section we present a semantics for well-named Interactive Generalized Semi-Markov Processes which maps them onto Interactive Stochastic Timed Transition Systems. Such a semantics explicitly represents the passage of time by means of transitions labeled with numeric time delays and turns probability distributions of durations into infinitely branching probabilistic choices which lead to states performing numeric time delays with a different duration. In particular, differently from [18] where a technique based on *residual lifetimes of clocks* is used, the states of the semantics of an Interactive Generalized Semi-Markov Process encode the *spent lifetimes of clocks*. This means that, in a timed state of the IGSMP where several clocks $C_{n_1} \dots C_{n_k}$ are in execution, the time delay originated by a clock C_{n_i} is determined according to its residual distribution of duration which is evaluated from (i) its associated duration distribution and (ii) its spent lifetime. Once we have sampled a time value t_i from the residual duration distribution of each clock C_{n_i}, we just take the minimum t_{min} of the sampled values and we consider the clock $C_{n_{\mathrm{min}}}$ which sampled such a time value. Such a "winning clock" is the clock that terminates in the timed state of the IGSMP. After this event the other clocks (which are still in execution) carry over their spent lifetimes, which now is given by $t'_i = t_i + t_{\mathrm{min}}$. Since, according to this approach, the residual duration of a clock is re-sampled in each IGSMP state until it terminates, an adversary (or scheduler) which resolves non-deterministic choices in an IGSMP cannot gain information about the future behavior of the system on which to base its decisions.

Example 5. Let us consider the IGSMP depicted in Fig. 13, where three temporal delays are started by activating three clocks C_{n_1}, C_{n_2}, and C_{n_3}. In particular, we concentrate on the case in which C_{n_2} is the first clock to terminate.

In Fig. 14 we show the semantics of the IGSMP of Fig. 13 obtained by following an approach similar to that of [18], which encodes in each state the *residual lifetimes of clocks*. Each state is enriched with the set of active clocks together with their residual lifetimes. In state $\langle s_0, \emptyset \rangle$ (where no clock is active) three numeric time delays t_1, t_2, and t_3 are sampled and associated with the lifetime of the clocks C_{n_1}, C_{n_2}, and C_{n_3}, respectively. Depending on which is the clock $C_{n_{\mathrm{min}}}$ sampling the minimum time value t_{min} in $\langle s_0, \emptyset \rangle$, we move to one of three different classes of states, one for each possible winning clock. Afterwards, a temporal transition labeled with a numeric time value t between 0 and t_{min} is taken, and each residual duration is accordingly modified by subtracting t_{min} from the residual lifetime of each clock. For the sake of readability in Fig. 14 we just depict one trace leading from s_0 to a state s_1 which belongs to the class of states for which C_{n_2} is the winning clock (i.e. t_2 is t_{min}), and then from s_1 to s_2 via the transition labeled with the time value t_2, so that in s_2 the clock C_{n_2} is terminated. In s_2 the residual lifetimes of the remaining active clocks C_{n_1} and C_{n_3} are $t_1 - t_{min}$ and $t_3 - t_{min}$ respectively. By exploiting this information an adversary may already know which clock between C_{n_1} and C_{n_3} will terminate first and consequently guide the nondeterministic choice in s_2.

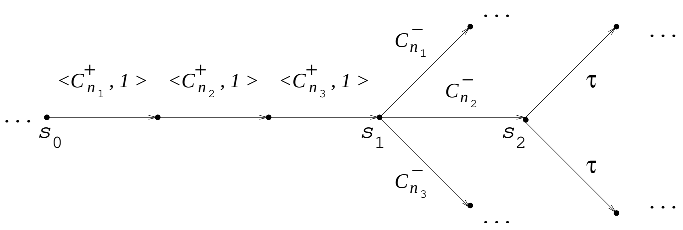


Fig. 13. Example of an IGSMP

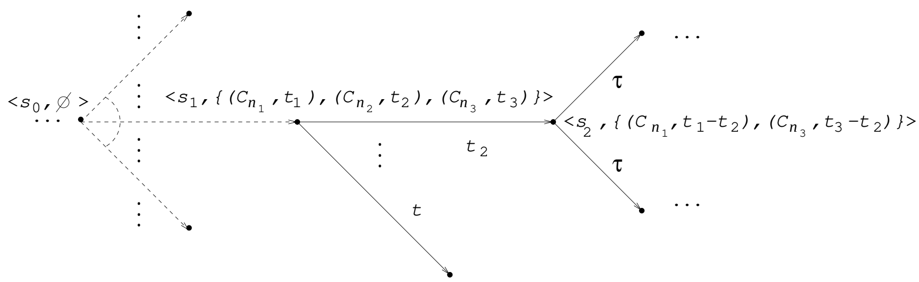


Fig. 14. Example of semantics based on residual lifetimes

In Fig. 15 we show the semantics of the IGSMP of Fig. 13 obtained by following the approach that we adopt in this paper, which is based on the *spent lifetimes of clocks*. Each state is enriched with: (i) the set of active clocks together with their spent lifetimes, and (ii) a pair $C_n : t$ containing the time value sampled by the winning clock in a timed state of the IGSMP and the clock name. The latter field is set to "$-$" whenever the IGSMP is not in a timed state. The sampling executed in $\langle s_0, \emptyset, -\rangle$ leads to a state where the three starting clocks are associated with the spent lifetime 0 (because the corresponding transition does not represent a passage of time but simply the result of the sampling), and the winning clock C_n and its sampled value are reported too. As in the case of Fig. 14, in Fig. 15 we just report one trace leading from s_0 to a state s_1 which belongs to the class of states for which C_{n2} is the winning clock (i.e. C_{n2} is $C_{n_{min}}$ and t_2 is its sampled value), and then from s_1 to s_2 via the transition labeled with the value t_2, so that in s_2 the clock C_{n_2} is terminated. In state s_2 the spent lifetimes of the remaining active clocks C_{n_1} and C_{n3} are both equal to t_2, and their residual durations depend on both such a value and the duration distribution associated with the clocks. Since, according to this approach, the time to termination of clocks C_{n_1} and C_{n3} is re-sampled, an adversary cannot gain in advance any information about the future behavior of the system and he cannot exploit this information when resolving the nondeterministic choice in s_2.

In the following we introduce some preliminary definitions which are needed to define the semantics of IGSMPs.

Definition 18. Given a duration probability distribution $f \in PDF^+$ and a time

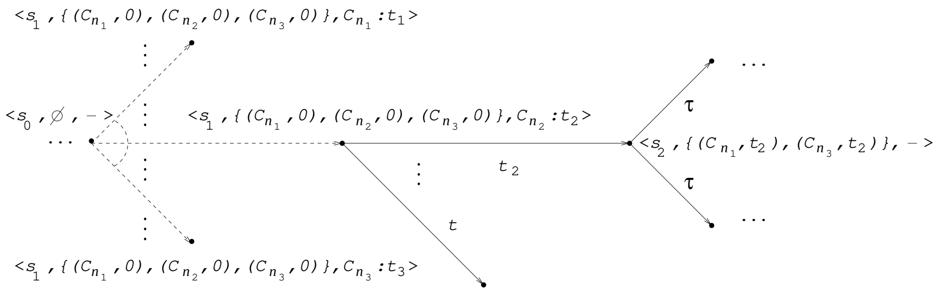


Fig. 15. Example of semantics based on spent lifetimes

value $t \in \mathbb{R}^+$, we denote by $[f \mid t]$ the residual duration distribution of a clock C_n with duration distribution f which, after t time units from when it started, has not terminated yet (t is its spent lifetime). More formally, if T is a random variable with distribution f, i.e. $\forall t' \in \mathbb{R}.\ f(t') = P(T \leq t')$, then $[f \mid t]$ is the probability distribution defined as follows. For all $t' \in \mathbb{R}$ we have that:

$$[f \mid t](t') = P(T \leq t' + t \mid T > t)$$ ■

Theorem 19. *Given* $f \in PDF^+$ *and* $t \in \mathbb{R}^+$*, we have that for all* $t' \in \mathbb{R}^+ \cup \{0\}$*:*

$$[f \mid t](t') = \frac{f(t+t') - f(t)}{1 - f(t)}$$ ■

Consider a family of probability distribution functions $f_1, \ldots, f_k \in PDF$. We denote by $\mathcal{R}(f_1, \ldots, f_k)$ the corresponding probability space over the domain $\mathbb{R}^k$ containing tuples of real values that can be sampled by $f_1, \ldots, f_k$ (sample space).

Definition 20. Let the residual duration distribution of the set of clocks C_{n_1}, $\ldots, C_{n_k}$ in execution in an IGSMP state be $f_1, \ldots, f_k$, i.e. the probability that a certain tuple of residual durations $(t_1, \ldots, t_k)$ is sampled from the clocks is described by the probability space $\mathcal{R}(f_1, \ldots, f_k)$. For each $I \subseteq \{1, \ldots, k\}$ such that $|I| \geq 2$, the event $Term(I)$ of contemporaneous termination of the clocks $\{C_{n_i} \mid i \in I\}$ in execution is the following subset of the sample space $\mathbb{R}^k$:

$$Term(I) = \{(t_1, \ldots, t_k) \mid \exists t.\ (\forall i \in I.\ t_i = t) \wedge (\forall i \notin I.\ t_i > t\}$$ ■

Since in an IGSMP clocks in execution in a state cannot terminate at the same time instant we have that each event $Term(I)$ of contemporaneous termination of a subset $\{C_{n_i} \mid i \in I\}$ of the clocks in execution $C_{n_1}, \ldots, C_{n_k}$ occurs with probability 0. We exploit this fact in order to reduce the domain of the probability space for a set of active clocks. In particular instead of considering the entire $\mathcal{R}(f_1, \ldots, f_k)$ we can just restrict to consider $\breve{\mathcal{R}}(f_1, \ldots, f_k)$ whose domain is $\breve{\mathbb{R}}^k = \mathbb{R}^k - \bigcup_{I \subseteq \{1, \ldots, k\}, |I| \geq 2} Term(I)$.

We are now in a position to formally define the semantics of an IGSMP.

$$\textbf{(P1)}\quad \frac{(\exists C_n.\ s \xrightarrow{C_n^-}) \ \wedge\ \{C_{n_1},\dots,C_{n_k}\} = dom(v)}{P(\langle s,v,-\rangle) = Sample^{\{\mathbf{n_i}\}}_{s,v}(\ \check{\mathcal{R}}([D(C_{n_1}) \mid v(C_{n_1})],\dots,[D(C_{n_k}) \mid v(C_{n_k})])\ )}$$

$$\textbf{(P2)}\quad \frac{(\exists C_n, w.\ s \xrightarrow{<C_n^+,w>}\ ) \ \wedge\ Pr = \{\ (\ <C_n, s'>,\ w/TW(s)\ )\ \mid\ s \xrightarrow{<C_n^+,w>} s'\ \}}{P(\langle s,v,-\rangle) = \sum_{<C_n,s'> \in dom(Pr)} Pr(<C_n,s'>) \cdot P(\langle s', v \cup \{(C_n,0)\}, -\rangle)}$$

$$\textbf{(T1)}\quad \langle s,v,C_n\!:\!t\rangle \xrightarrow{t'} \langle s, v+t', -\rangle \qquad 0 \le t' < t$$

$$\textbf{(T2)}\quad \frac{s \xrightarrow{C_n^-} s'}{\langle s,v,C_n\!:\!t\rangle \xrightarrow{t} \langle s', (v - C_n) + t, -\rangle}$$

$$\textbf{(T3)}\quad \frac{(\nexists\theta.\ s \xrightarrow{\theta}) \ \wedge\ s \not\xrightarrow{\tau}}{\langle s,\emptyset,-\rangle \xrightarrow{t} \langle s,\emptyset,-\rangle} \qquad t \ge 0$$

$$\textbf{(A1)}\quad \frac{s \xrightarrow{\alpha} s'}{\langle s,v,-\rangle \xrightarrow{\alpha} \langle s',v,-\rangle} \qquad\qquad \textbf{(A2)}\quad \frac{s \xrightarrow{a} s'}{\langle s,v,C_n\!:\!t\rangle \xrightarrow{a} \langle s',v,-\rangle}$$

$$TW(s) = \sum \{\!|\ w \mid \exists C_n.\ s \xrightarrow{<C_n^+,w>}\ |\!\}$$

$$Sample^{\{\mathbf{n_i}\}}_{s,v}(t_1,\dots,t_k) = \langle s, v, C_{n_{min}}\!:\!t_{min}\rangle$$

where min is the only index i such that: $\mathrm{t_i} = \min_{\mathrm{j} \in \{1,\dots,\mathrm{k}\}} \mathrm{t_j}$

Table 2. Semantic rules for IGSMPs

Definition 21. The semantics of an IGSMP $\mathcal{G} = (\Sigma, \mathcal{C}, D, Act, T_+, T_-, T_a, s_0)$ is the RISTTS $[\![\mathcal{G}]\!] = (\Sigma', \mathbb{R}^+ \cup \{0\}, Act, P, T_t, T'_a, s'_0)$ where:

- $\Sigma' = (\Sigma \times \mathit{Spent} \times \mathit{Sample})$ is the set of states of the RISTTS, where *Spent*, ranged over by v, is the set of partial functions from $\mathcal{C}$ to $\mathbb{R}^+ \cup \{0\}$, expressing the time already spent in execution by the clocks currently in execution in the IGSMP (clocks in the domain of *Spent*), and *Sample*, ranged over by *sample*, is the set $(\mathcal{C} \times (\mathbb{R}^+ \cup \{0\})\,) \cup \{-\}$, where a pair (C_n, t), also written $C_n : t$, denotes that the IGSMP is currently executing a set of clocks and that clock C_n has sampled the minimum residual time delay with t being the value of such a delay; while "$-$" denotes that started clocks are not under execution (e.g. the IGSMP is in a choice state or in a silent state).
- $\mathbb{R}^+ \cup \{0\}$ is the time domain: we consider continuous time.

- Act is the set of standard actions considered in the IGSMP.
- P, which associates a probability space (expressing next state probability) to some of the states in Σ', is defined to be the least partial function on Σ' satisfying the operational rules in the first part of Table 2.
- T_t is the set of timed transitions which are defined as the least relation over $\Sigma' \times (\mathbb{R}^+ \cup \{0\}) \times \Sigma'$ satisfying the operational rules in the second part of Table 2.
- T'_a is the set of action transitions which are defined as the least relation over $\Sigma' \times Act \times \Sigma'$ satisfying the operational rules in the third part of Table 2.
- $s'_0 = \langle s_0, \emptyset, - \rangle$ is the initial state of the RISTTS, where the IGSMP is in the initial state and no clock is in execution. ■

In Table 2 we make use of the following notation. Given $v \in Spent$, we define $v - C_n$ to be the partial function obtained from v by removing C_n (and the associated value) from its domain. We define $v + t$, with $t \in \mathbb{R}^+ \cup 0$, to be the partial function obtained from v by adding t to the time value associated with each clock in the domain of v. We use the notation $\{\mathbf{n_i}\}$ to stand for $\{\mathbf{n_i}\}_{\mathbf{i=1\ldots k}}$, representing the sequence of names $n_1, \ldots n_k$ (in Table 2 the length k of the sequence is always clarified by the context in which $\{\mathbf{n_i}\}$ is used). Finally in the fourth part of Table 2 we define two auxiliary functions. The function $TW : \Sigma \longrightarrow \mathbb{R}^+ \cup \{0\}$ computes the overall weight of the clock start transitions leaving a state of an IGSMP. Moreover, given a state of the IGSMP $s \in \Sigma$, a partial function mapping active clock into their spent lifetimes $v \in Spent$, and a sequence $\{n_1, \ldots, n_k\}$ of clock indexes, the function $Sample^{\{\mathbf{n_i}\}}_{s,v}$ maps a tuple $(t_1, \ldots, t_k)$ of time values sampled by clocks into the corresponding state $\langle s, v, C_{n_{\min}} : t_{\min} \rangle$ reached in the RISTTS, where min is the index of the clock which sampled the least time value. Note that function $Sample^{\{\mathbf{n_i}\}}_{s,v}$ is used in Table 2 for deriving (via induction) a probability space over the states of the RISTTS from the probability space $\breve{\mathcal{R}}([D(C_{n_1}) \mid v(C_{n_1})], \ldots, [D(C_{n_k}) \mid v(C_{n_k})])$ over residual durations sampled by active clocks in a state of the IGSMP. Finally, we assume that, for every distribution f, $[f|0]$ is defined so that it yields the distribution f itself.

The following theorem shows that the semantics of well-named IGSMPs preserves weak bisimulation and is indeed compositional.

Theorem 22. *Let $\mathcal{G}'$, $\mathcal{G}''$ be two well-named IGSMPs. It holds that:*

- *If $\mathcal{G}' \approx \mathcal{G}''$ then $[\![\mathcal{G}']\!] \approx [\![\mathcal{G}'']\!]$;*
- *For each $S \subseteq Act - \{\tau\}$ we have $[\![\mathcal{G}']\!] \parallel_S [\![\mathcal{G}'']\!] \approx [\![\mathcal{G}' \parallel_S \mathcal{G}'']\!]$;*
- *For each $L \subseteq Act - \{\tau\}$ we have $[\![\mathcal{G}']\!]/L \approx [\![\mathcal{G}'/L]\!]$.* ■

3 Calculus of Interactive Generalized Semi-Markov Processes

In this section we introduce the calculus of Interactive Generalized Semi-Markov Processes, a stochastic process algebra which can express probabilistic timed delays with general distributions and synchronizable actions with zero duration.

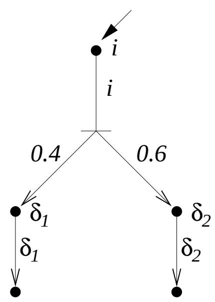
The calculus of *IGSMPs* is equipped with a structural operational semantics which generates semantic models in the form of well-named *IGSMPs*, the interactive extension of *GSMPs* presented in the previous Sect. 2. This is obtained by expressing the concurrent execution of delays through a simple probabilistic extension of ST semantics based on dynamic names [13]. We also show that the notion of observational congruence over *IGMSPs*, obtained by modifying weak bisimulation over *IGSMPs* (introduced in the previous Sect. 2) in the standard way [27], is a congruence for all the operators of the calculus over strongly guarded processes and we produce an axiomatization for this equivalence which is complete over finite-state strongly guarded processes.

This section is structured as follows. In Sect. 3.1 we present the basic concepts on which our approach is based. Then, in Sect. 3.2 we present the calculus of Interactive *GSMPs* and its operational semantics. Finally, in Sect. 3.3 we present the notion of observational congruence and in Sect. 3.4 its complete axiomatization.

3.1 Basic Concepts

The Basic Idea Previous work has been done in order to try to extend the expressiveness of Markovian process algebras to probabilistic time with general distributions (see e.g. [21, 2, 29]). The main point in doing this is to understand how to define the algebra operational semantics and semantic reasoning, e.g. the definition of an adequate notion of bisimulation based equivalence. Once recognized that we have to produce a system representation like that of the IGSMP model presented in Sect. 2 when dealing with general distributions, an important issue is how to develop a calculus suitable for generating IGSMPs, so to have a fully compositional approach. The problem of developing a semantics for a process algebra with generally distributed delays (expressed, e.g., by $f.\_$ prefixes, where f is a probability distribution) is essentially the problem of representing the execution of a temporal delay as the combination of a start and a termination event in such a way that pair of events derived from the execution of a delay are somehow related, e.g. by deriving a unique clock name for the delay (see Sect. 2). As we recognized in [12] such problem is not new in the literature, but exactly corresponds to consider classical *ST semantics* [20] for delays. With respect to the definition of ST semantics the "type" of a delay is simply its duration distribution f and what we observe of a system is its ability of performing delays of certain types f. In particular *identification* of delays by means of clock names is obtained if we use a technique for expressing ST semantics based on names (like the static and dynamic name techniques presented in [13]) and not, e.g., on pointers [15]. The use of such a semantics cause clock names to be generated for delays by the semantics, thus obtaining models with clocks like IGSMPs.

Another important issue to address when developing a semantics for a process algebra with generally distributed delays is how to interpret a choice "$\_ + \_$" between delays. Our claim (introduced in [11]) is that, while in the "Markovian world" the most natural solution is to adopt a *race policy* where the choice



$$PDF(\delta_1) = f, \quad PDF(\delta_2) = g, \quad PDF(i) = Det(0)$$

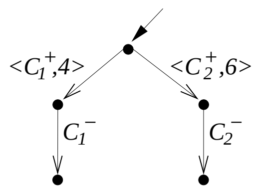
Fig. 16. Choice of Generally Distributed Delays

is solved in favor of the delay which terminates first, when we consider general distributions a convenient solution is to adopt a *preselection policy* where, first one of the delays is selected according to a probabilistic choice, and then the selected delay is executed. A justification for this claim can be found in the difference between the structure of GSMPs, where probabilistic choices are explicitly represented, and CTMCs, where probabilistic choices are implicitly expressed by races between exponential delays. For example an intuitive semantics of $f.\underline{0} + g.\underline{0}$, where f and g are generally distributed delays, should generate a GSMP like that of Fig. 16, where "i" is an auxiliary dummy element with zero duration which causes the process to leave immediately the first state. [7]

The GSMP of Fig. 16 first performs a probabilistic choice between f and g (e.g. according to probabilities 0.4 and 0.6) and then executes the selected delay. This corresponds to using the *preselection policy* instead of the race policy to solve choices. Obviously we must somehow syntactically express the probability associated with delays in a choice (in the GSMP above 0.4 and 0.6 for f and g, respectively). This can be done by using delay prefixes $<f, w>.\_$ in the algebra, so that the weight w determines the probability of selecting the delay. In this way the GSMP would be obtained from, e.g., $<f, 4>.\underline{0} + <g, 6>.\underline{0}$. Notably, such a preselection policy is naturally obtained from ST semantics applied to $<f, w>.\_$ prefixes by associating the weight w with the transition representing the delay start. For instance in the case of $<f, 4>.\underline{0} + <g, 6>.\underline{0}$ we would obtain the IGSMP of Fig. 17 whose performance model is the GSMP of Fig. 16.

In [21, 2, 29] efforts have been made in order to try to adapt the interleaving semantics of standard process algebras [27, 24] to deal with general distributions (similarly as already done in the simpler context of Markovian process algebras), instead of splitting delays in starting and terminating events as in the ST semantics. When doing this, the basic idea is considering actions as starting in the

[7] An equivalent representation of the GSMP of Fig. 16 which allows us to leave out the element i can be obtained by associating directly with the states labeled with δ_1 and δ_2 the probability of being the initial state.



$$PDF(C_1) = f, \qquad PDF(C_2) = g$$

Fig. 17. Event-Based Representation of Choice of Delays

first state they become enabled. In order for the starting point of actions to be observable the interleaving semantics had to be enriched with additional information: in [21] transitions are enriched with *start references*, in [29] transitions are enriched with information about causality relations among actions, and in [2] actions must be differently identified with indexes before the semantic rules are applied. As a matter of fact these semantics are not actually interleaving since, even if they consider choices to be resolved with race policy, the expansion law $a.\underline{0} \,||\, b.\underline{0} = a.b.\underline{0} + b.a.\underline{0}$ is no longer valid, due to the fact that distributions are not memoryless. Hence there is no advantage in trying to keep the semantics in an interleaving atomic form and to preserve the rather tricky race policy for the alternative composition operator "$\_ + \_$".

On the contrary, if we use ST semantics we also obtain an expansion law for generally distributed delays at the level of clock start and clock termination events which allow us to produce a complete axiomatization over finite-state algebraic specifications.

A Semantics for Concurrent Systems with General Distributions The process algebra that we will consider (called calculus of IGSMPs) is an extension of a standard process algebra [27, 24] with new prefixes "$<f, w>.\_$", representing timed delays with general probability duration distribution f and associated weight w, which produces well-named IGSMPs as semantic models of its terms.

As we already explained, we define the operational semantics of a delay f through a realization of ST semantics based on names. In [13] we have introduced two name based techniques for expressing ST semantics. With the first one ST semantics is expressed by assigning *static names* to delays, i.e. names determined according to their syntactical position in the system, while in the second one we use *dynamic names*, i.e. names computed dynamically while the system evolves. When defining the semantics of the calculus of IGSMPs, we use the dynamic name technique which, with respect to the static name technique, has the advantage of generating semantic models such that ST bisimulation can be simply established via the standard notion of observational congruence [27]. In particular, the use of the dynamic name technique allows us to derive semantic models in the form of well-named IGSMPs presented in Sect. 2, for which equivalence is defined as a simple extension of probabilistic bisimulation [25] (so that existing results and tools can be exploited). On the contrary, using static names for the semantics of generally distributed delays requires a more complex definition of bisimulation which associates the names of the delays of one process with the names of the corresponding delays used by the other one.

A delay is represented in semantic models as a combination of the event of start of the delay f^+ and the event of termination of the delay f^-. Moreover, according to the dynamic name technique, we dynamically assign indexes i to delays so that the execution of a delay is represented by the two events f_i^+ and f_i^- and no confusion arises (in the connection between delay starts and delay terminations) when multiple delays with the same distribution f are concurrently executed. We resolve choices among several delays by means of preselection policy. In a choice a delay is selected with probability proportional to its associated weight. For instance $<f,w>.\underline{0} + <g,w'>.\underline{0}$ represents a system which performs a delay of distribution f with probability $w/(w+w')$ and a delay of distribution g with probability $w'/(w+w')$. Choices are expressed in semantic models by associating weights to transitions f_i^+ representing the start of a delay. With respect to the definition of well-named IGSMPs of Sect. 2, in the rest of the paper we will use "$<f_i^+,w>$" and "f_i^-" as shorthands for "$<C_{f,i}^+,w>$" and "$C_{f,i}^-$", respectively, in transition labels.

The semantics of standard actions a (including internal τ) in *IGSMP* is, instead, just the standard interleaving semantics. This reflects the fact that these actions have zero duration and can be considered as being executed atomically.

In general, the semantics of terms of the calculus of IGSMPs is simply defined via SOS by extending the standard interleaving semantics for action prefixes [27] with ST semantics via dynamic names for delay prefixes "$<f,w>$". This can be done easily with the technique introduced in [13] which is compositional. Moreover, the interrelation among the three kind of derived transitions is captured in the semantics by: (i) applying a notion of priority of delay start transitions over delay termination transitions due to the assumption that probabilistic choices are resolved urgently, and (ii) applying a notion of priority of τ transitions over time related transitions due to the maximal progress assumption. We also show that the notion of observational congruence over terms of the calculus, obtained by modifying weak bisimulation over well-named *IGSMPs* defined in Sect. 2.2 in the standard way [27], is a congruence for all the operators of the calculus over strongly guarded processes and we produce an axiomatization for this equivalence which is complete over finite-state strongly guarded processes.

3.2 The Calculus of Interactive GSMPs

Syntax of Terms and Informal Semantics of Operators The calculus of interactive *GSMPs* is an extension of a standard process algebra with operators of CCS/CSP [27, 24] which allows us to express priority, probabilistic choices and probabilistic delays with arbitrary distributions. This is done by including into the calculus, in addition to standard actions, a special kind of actions representing delays. Delays are represented as $<f,w>$ and are characterized by a *weight* w and a *duration distribution* f. The weight $w \in \mathbb{R}^+$ determines the probability of choosing the delay in a choice among several delays. The duration distribution $f \in PDF^+$ denotes the probability distribution function of the delay duration. The possibility of expressing priority derives from the interrelation of delays and standard τ actions. In particular we make the maximal progress assumption: the system cannot wait if it has something internal to do. Therefore we assume that, in a choice, τ actions have priority over delays, i.e. $\tau.P + {<}f, w{>}.Q$ behaves as $\tau.P$.

Let Act be the set of action types as in Sect. 2.2; $TAct = \{{<}f, w{>} \mid f \in PDF^+ \wedge w \in \mathbb{R}^+\}$ be the set of delays. [8]; Var be a set of process variables ranged over by X, Y, Z; and $ARFun = \{\varphi : Act \longrightarrow Act \mid \varphi(\tau) = \tau \wedge \varphi(Act - \{\tau\}) \subseteq Act - \{\tau\}\}$ be a set of *action relabeling functions*, ranged over by φ.

Definition 23. We define the language *IGSMP* as the set of terms generated by:

$$P ::= \underline{0} \mid X \mid {<}f, w{>}.P \mid \alpha.P \mid P + P \mid P/L \mid P[\varphi] \mid P \parallel_S P \mid recX.P$$

where $L, S \subseteq Act - \{\tau\}$. An *IGSMP* process is a closed term of *IGSMP*. We denote by $IGSMP_g$ the set of strongly guarded terms of *IGSMP*. [9] ∎

"$\underline{0}$" denotes a process that cannot move. The operators "." and "+" are the CCS prefix and choice. The choice among delays is carried out through the preselection policy by giving each of them a probability proportional to its weight. Note that alternative delays are not executed concurrently, first one of them is chosen probabilistically and then the selected delay is executed. Moreover τ actions have priority over delays in a choice. "$/L$" is the hiding operator which turns into τ the actions in L, "$[\varphi]$" is the relabeling operator which relabels visible actions according to φ. "$\parallel_S$" is the CSP parallel operator, where synchronization over actions in S is required. Finally "$recX$" denotes recursion in the usual way.

In this paper we will just consider terms of $IGSMP_g$, i.e. strongly guarded terms of our calculus. Weakly guarded processes can be dealt with by using an approach like that introduced in [5, 7].

Operational Semantics As explained in Sect. 3.1, we employ the dynamic name technique of [13] for giving semantics to generally distributed delays f.

The problem of preserving the relationship between starts and terminations of delays arises, like in the ST semantics of standard process algebras, when several delays of the same type f are being executed in parallel. When a delay f terminates (event f^-) we need some information for establishing which event of delay start (f^+) it refers to. By using the dynamic name technique introduced in [13] this is obtained by generating a fresh name f_i for each starting delay f, where $i \in \mathbb{N}$ is the minimum index not already used by the other delays with distribution f that have started but not yet terminated, thus obtaining names for clocks in the format required by well-named IGSMPs. In particular, due to the

[8] In the following we consider f to be a shorthand for ${<}f, 1{>}$ when this is clear from the context.

[9] We consider the delay ${<}f, w{>}$ as being a guard in the definition of strong guardedness.

adoption of preselection policy (see Sect. 3.1) starting events generated by delay prefixes $<f, w>$ are represented in semantic models by a transitions labeled by $<f_i^+, w>$, where i is determined as explained above. The termination of a delay prefix $<f, w>$ is simply represented by a transition labeled by f_i^-, where the "identifier" i uniquely determines which delay f is terminating.

Moreover the dynamic name technique introduced in [13] allows us to dynamically assign names to delays, according to the rule formerly described, via SOS semantics (hence in a *compositional way*) through the idea of *levelwise renaming*. In order to obtain structural compositionality it is necessary to determine at each syntactical level of a term, e.g. in the case of the parallel composition operator, the computations of $P \,\|\, Q$ from the computations of P and Q. This is done, as described in Sect. 2.2 when composing in parallel well-named IGSMPs, by suitably renaming delays occurring in the computations of P and Q. In particular, in the operational semantics the mapping $M \in \mathcal{M}$ recording delay renamings for a certain parallel operator (see Sect. 2.2) is expressed as an additional parameter of that parallel operator in state terms. For every delay f started by $P \,\|_{S,M}\, Q$, M records the association between the name f_i, generated according to the well-naming rule for identifying f at the level of $P \,\|_{S,M}\, Q$, and the name f_j (which in general is different from f_i), generated according to the well-naming rule for identifying the same delay f inside P (or Q). In this way when, afterwards, such a delay f terminates in P (or Q) the name f_j can be re-mapped to the correct name f_i at the level of $P \,\|_{S,M}\, Q$, by exploiting the information included in M. As explained in Sect. 2.2, in M the delay f of $P \,\|_{S,M}\, Q$ which gets index i is uniquely identified by expressing the unique name j it gets in P or in Q and the "location" of the process that executes it: *left* if P, *right* if Q. Such an association is represented inside M by the triple $f : (i, loc_j)$.

In order to define the operational semantics for the processes of $IGSMP$, we need a richer syntax to represent states. Let $TAct^+ = \{<f_i^+, w> \mid f \in PDF^+ \wedge i \in \mathbb{N}^+ \wedge w \in \mathbb{R}^+\}$ be the set of delay starts, where $<f_i^+, w>$ represents the beginning of the delay $<f, w>$ identified by i. [10] Besides let $TAct^- = \{f_i^- \mid f \in PDF^+ \wedge i \in \mathbb{N}^+\}$ be the set of delay terminations, where f_i^-, represents the termination of the delay with duration distribution f identified by i. η ranges over $Act \cup TAct \cup TAct^+ \cup TAct^-$.

The set $IGSMP_s$ of state terms of $IGSMP$ is generated by:

$$P ::= \underline{0} \mid X \mid \eta.P \mid P + P \mid P/L \mid P[\varphi] \mid P \,\|_{S,M}\, P \mid recX.P$$

We denote by $IGSMP_{sg}$ the set of strongly guarded terms of $IGSMP_s$. We consider the operators "$\|_S$" occurring in a $IGSMP$ term P as being "$\|_{S,\emptyset}$" when P is regarded as a state.

The semantics of state terms produces a transition system labeled over $Act \cup TAct^+ \cup TAct^-$, ranged over by $\gamma, \gamma', \ldots$. Such a transition system is defined as being the well-named $IGSMTS$ [11] $\mathcal{G} = (IGSMP_{sg}, Act, T_+, T_-, T_a)$, where: T_a is

[10] In the following we consider f_i^+ to be a shorthand for $<f_i^+, 1>$ when this is clear from the context.

[11] We recall that we consider f_i^+ and f_i^- as being shorthands for $C_{f,i}^+$ and $C_{f,i}^-$ in the transitions of the IGSMTS.

$$\alpha.P \xrightarrow{\alpha} P$$

$$\frac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'} \qquad \frac{Q \xrightarrow{\alpha} Q'}{P + Q \xrightarrow{\alpha} Q'}$$

$$\frac{P \xrightarrow{\alpha} P'}{P \parallel_{S,M} Q \xrightarrow{\alpha} P' \parallel_{S,M} Q} \ \alpha \notin S \qquad \frac{Q \xrightarrow{\alpha} Q'}{P \parallel_{S,M} Q \xrightarrow{\alpha} P \parallel_{S,M} Q'} \ \alpha \notin S$$

$$\frac{P \xrightarrow{a} P' \quad Q \xrightarrow{a} Q'}{P \parallel_{S,M} Q \xrightarrow{a} P' \parallel_{S,M} Q'} \ a \in S$$

$$\frac{P \xrightarrow{a} P'}{P/L \xrightarrow{\tau} P'/L} \ a \in L \qquad \frac{P \xrightarrow{\alpha} P'}{P/L \xrightarrow{\alpha} P'/L} \ a \notin L$$

$$\frac{P \xrightarrow{\alpha} P'}{P[\varphi] \xrightarrow{\varphi(\alpha)} P'[\varphi]} \qquad \frac{P\{recX.P/X\} \xrightarrow{\alpha} P'}{recX.P \xrightarrow{\alpha} P'}$$

Table 3. Standard Rules

$$<f,w>.P \xrightarrow{<f_i^+,w>} f_i^-.P \qquad <f_i^+,w>.P \xrightarrow{<f_i^+,w>} P$$

$$\frac{P \xrightarrow{<f_i^+,w>} P' \ \wedge \ Q \stackrel{\tau}{\not\longrightarrow}}{P + Q \xrightarrow{<f_i^+,w>} P'} \qquad \frac{Q \xrightarrow{<f_i^+,w>} Q' \ \wedge \ P \stackrel{\tau}{\not\longrightarrow}}{P + Q \xrightarrow{<f_i^+,w>} Q'}$$

$$\frac{P \xrightarrow{<f_i^+,w>} P' \ \wedge \ Q \stackrel{\tau}{\not\longrightarrow}}{P \parallel_{S,M} Q \xrightarrow{<f_{n(M_f)}^+,w>} P' \parallel_{S,M\cup\{f:(n(M_f),l_i)\}} Q}$$

$$\frac{Q \xrightarrow{<f_i^+,w>} Q' \ \wedge \ P \stackrel{\tau}{\not\longrightarrow}}{P \parallel_{S,M} Q \xrightarrow{<f_{n(M_f)}^+,w>} P \parallel_{S,M\cup\{f:(n(M_f),r_i)\}} Q'}$$

$$\frac{P \xrightarrow{<f_i^+,w>} P' \wedge \not\exists a \in L.\, P \xrightarrow{a}}{P/L \xrightarrow{<f_i^+,w>} P'/L} \qquad \frac{P \xrightarrow{<f_i^+,w>} P'}{P[\varphi] \xrightarrow{<f_i^+,w>} P'[\varphi]}$$

$$\frac{P\{recX.P/X\} \xrightarrow{<f_i^+,w>} P'}{recX.P \xrightarrow{<f_i^+,w>} P'}$$

Table 4. Rules for Start Moves

$$f_i^-.P \xrightarrow{f_i^-} P$$

$$\frac{P \xrightarrow{f_i^-} P' \,\wedge\, Q \not\xrightarrow{\tau} \,\wedge\, Q \not\xrightarrow{<f_k^+,w>}}{P + Q \xrightarrow{f_i^-} P'} \qquad \frac{Q \xrightarrow{f_i^-} Q' \,\wedge\, P \not\xrightarrow{\tau} \,\wedge\, P \not\xrightarrow{<f_k^+,w>}}{P + Q \xrightarrow{f_i^-} Q'}$$

$$\frac{P \xrightarrow{f_i^-} P' \,\wedge\, Q \not\xrightarrow{\tau} \,\wedge\, Q \not\xrightarrow{<f_k^+,w>}}{P \,\|_{S,M}\, Q \xrightarrow{f_j^-} P' \,\|_{S,M-\{f:(j,l_i)\}}\, Q} \quad f:(j,l_i) \in M$$

$$\frac{Q \xrightarrow{f_i^-} Q' \,\wedge\, P \not\xrightarrow{\tau} \,\wedge\, P \not\xrightarrow{<f_k^+,w>}}{P \,\|_{S,M}\, Q \xrightarrow{f_j^-} P \,\|_{S,M-\{f:(j,r_i)\}}\, Q'} \quad f:(j,r_i) \in M$$

$$\frac{P \xrightarrow{f_i^-} P' \wedge \nexists a \in L.\, P \xrightarrow{a}}{P/L \xrightarrow{f_i^-} P'/L} \qquad \frac{P \xrightarrow{f_i^-} P'}{P[\varphi] \xrightarrow{f_i^-} P'[\varphi]}$$

$$\frac{P\{recX.P/X\} \xrightarrow{f_i^-} P'}{recX.P \xrightarrow{f_i^-} P'}$$

Table 5. Rules for Termination Moves

the least subset of $IGSMP_{sg} \times Act \times IGSMP_{sg}$ satisfying the standard operational rules of Table 3, T_+ is obtained from the least multiset over $IGSMP_{sg} \times TAct^+ \times IGSMP_{sg}$ satisfying the operational rules of Table 4 (similarly to [23], we consider a transition to have arity m if and only if it can be derived in m possible ways from the operational rules) by summing the weights of the multiple occurrences of the same transition, and T_- is the least subset of $IGSMP_{sg} \times TAct^- \times IGSMP_{sg}$ satisfying the operational rules of Table 5. In Tables 4 and 5 we use $P \xrightarrow{a}$ to stand for $\exists P' : P \xrightarrow{a} P'$, $P \not\xrightarrow{\tau}$ to stand for $\nexists Q : P \xrightarrow{\tau} Q$ and $P \not\xrightarrow{<f_k^+,w>}$ to stand for $\nexists f, w, k, Q : P \xrightarrow{<f_k^+,w>} Q$.

The rules of Table 4 define the transitions representing the start of a delay, by taking into account the priority of "τ" actions over delays. In particular the rules for parallel composition and hiding are exactly as those defined in Sect 2.2.

The rules of Table 5 define the transitions representing the termination of a delay, by taking into account the priority of "τ" actions over delay related transitions and the priority of delay starts over delay terminations. In particular the rules for parallel composition and hiding are exactly as those defined in Sect. 2.2.

Note that even if the operational rules in Tables 4 and 5 include negative premises, this does not cause inconsistencies because when applying such rules

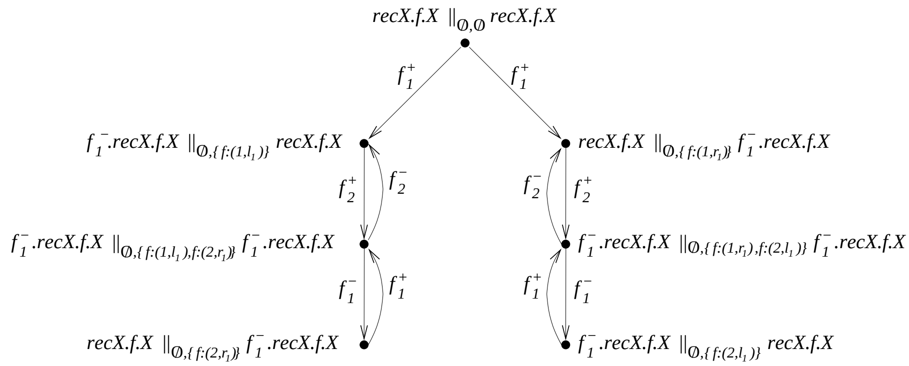


Fig. 18. Example of recursive system

for deriving the moves of a term P, the negative premises always refer to the moves of a subterm of P (and not of P itself), hence the operational semantics is stratifiable.

We are now in a position to define the well-named *IGSMP* obtained as the semantic model of a process of the calculus.

Definition 24. The semantic model $\mathcal{G}[\![P]\!]$ of $P \in \mathit{IGSMP}_g$ is the well-named *IGSMP* defined by $\mathcal{G}[\![P]\!] = (S_P, Act, T_{+,P}, T_{-,P}, T_{a,P}, P)$, where:

- S_P is the least subset of IGSMP_{sg} such that:
  - $P \in S_P$
  - if $P' \in S_P$ and $P' \xrightarrow{\gamma} P''$, then $P'' \in S_P$
- $T_{+,P}, T_{-,P}$ and $T_{a,P}$ are the restriction of T_+, T_- and T_a to $S_P \times Act \times S_P$, $S_P \times \mathit{TAct}^+ \times S_P$ and $S_P \times \mathit{TAct}^- \times S_P$. ■

Example 6. In Fig. 18 we depict the semantic model of $recX.f.X \,\|_\emptyset\, recX.f.X$. As expected, we obtain the same well-named IGSMP as that derived in Sect. 2 via parallel composition of two well-named IGSMPs that repeatedly execute f delays (see Fig. 9). ■

In the following theorem, where we consider "P/L", "$P[\varphi]$", and "$P \,\|_S\, P$" to be *static* operators [27], we show that finite semantic models are obtained for a wide class of recursive systems.

Theorem 25. *Let P be a IGSMP_g process such that for each subterm $recX.Q$ of P, X does not occur free in Q in the context of a static operator. Then P is a finite state process.*

Note that the class of processes considered in this corollary includes strictly the class of *nets of automata*, i.e. terms where no static operator occurs in the scope of any recursion.

3.3 Observational Congruence for IGSMP

In this section we introduce the notion of observational congruence over $IGSMP_{sg}$ processes, and we will show it to be a congruence with respect to all the operators of our calculus. Such a notion is defined, starting from the notion of weak bisimulation over Well-Named IGSMPs we defined in Sect. 2.2, according to the classical notion of observational congruence [27] and probabilistic bisimulation [25]. In the definition of observational congruence we make use of function TW defined in Sect. 2.2.

Definition 26. Two closed terms P, Q of $IGSMP_{sg}$ are observational congruent, written $P \simeq Q$, iff:

- for every $\sigma \in NPAct$ and $P' \in IGSMP_{sg}$,
  $P \xrightarrow{\sigma} P'$ implies $Q \stackrel{\sigma}{\Longrightarrow} Q'$ for some Q' with $P' \approx Q'$,
- for every $\sigma \in NPAct$ and $Q' \in IGSMP_{sg}$,
  $Q \xrightarrow{\sigma} Q'$ implies $P \stackrel{\sigma}{\Longrightarrow} P'$ for some P' with $P' \approx Q'$,
- for every $f \in PDF^+$ and equivalence class I of β,
  $TW(P, f, I) = TW(Q, f, I)$ ■

We consider $\simeq$ as being defined also on the open terms of $IGSMP_{sg}$ by extending observational congruence with the standard approach of [27].

Theorem 27. *$\simeq$ is a congruence w.r.t. all the operators of IGSMP, including recursion.*

3.4 Axiomatization

In this section we present an axiom system which is complete for $\simeq$ on finite state $IGSMP_{sg}$ terms.

The axiom system $\mathcal{A}_{IGSMP}$ for $\simeq$ on $IGSMP_{sg}$ terms is formed by the axioms presented in Fig. 19. In this figure "$\lfloor\!\lfloor$" and "$|$" denote, respectively, the left merge and synchronization merge operators. Moreover θ ranges over $TAct^+ \cup TAct^-$. We recall from Sect. 3.2 that $\gamma, \gamma', \ldots$ range over $Act \cup TAct^+ \cup TAct^-$.

The axioms $(Pri1)$ and $(Pri2)$ express the two kinds of priorities of $IGSMP$, respectively, priority of τ actions over (semi-)delays and priority of delay starts over delay terminations. The axiom (Par) is the standard one except that when the position of processes P and Q is exchanged we must invert left and right inside M. The inverse $\overline{M}$ of a mapping M is defined by $\overline{M} = \{f : (i, r_j) \mid f : (i, l_j) \in M\} \cup \{f : (i, l_j) \mid f : (i, r_j) \in M\}$. Axioms $(LM4)$ and $(LM5)$ just reflect the operational rules of the parallel operator for a delay move of the left-hand process. The axioms $(Rec1-3)$ handle strongly guarded recursion in the standard way [27].

If we consider the obvious operational rules for "$\lfloor\!\lfloor_{S,M}$" and "$|_{S,M}$" that derive from those we presented for the parallel operator [12] then the axioms of $\mathcal{A}_{IGSMP}$ are sound.

[12] The definition of the operational rule for "$|_{S,M}$" must allow for actions "τ" to be skipped [1], as reflected by axiom $(SM4)$.

$(A1)$	$P + Q = Q + P$	$(A2)$	$(P + Q) + R = P + (Q + R)$
$(A3)$	$\alpha.P + \alpha.P = \alpha.P$	$(A4)$	$P + \underline{0} = P$
$(Tau1)$	$\gamma.\tau.P = \gamma.P$	$(Tau2)$	$P + \tau.P = \tau.P$
$(Tau3)$	$\alpha.(P + \tau.Q) + \alpha.Q = \alpha.(P + \tau.Q)$		
$(TAct)$	$<f, w>.P = <f_1^+, w>.f_1^-.P$		
$(Prob)$	$<f_i^+, w>.P + <f_i^+, w'>.P = <f_i^+, w + w'>.P$		
$(Pri1)$	$\tau.P + \theta.Q = \tau.P$	$(Pri2)$	$<f_i^+, w>.P + g_j^-.Q = <f_i^+, w>.P$
$(Hi1)$	$\underline{0}/L = \underline{0}$	$(Hi2)$	$(\gamma.P)/L = \gamma.(P/L) \qquad \gamma \notin L$
$(Hi3)$	$(a.P)/L = \tau.(P/L) \qquad a \in L$	$(Hi4)$	$(P + Q)/L = P/L + Q/L$
$(Rel1)$	$\underline{0}[\varphi] = \underline{0}$	$(Rel2)$	$(\alpha.P)[\varphi] = \varphi(\alpha).(P[\varphi])$
$(Rel3)$	$(\theta.P)[\varphi] = \theta.(P[\varphi])$	$(Rel4)$	$(P + Q)[\varphi] = P[\varphi] + Q[\varphi]$
(Par)	$P \parallel_{S,M} Q = P \lfloor\!\lfloor_{S,M} Q + Q \lfloor\!\lfloor_{S,\overline{M}} P + P \mid_{S,M} Q$		
$(LM1)$	$\underline{0} \lfloor\!\lfloor_{S,M} P = \underline{0}$		
$(LM2)$	$(a.P) \lfloor\!\lfloor_{S,M} Q = \underline{0}$		$a \in S$
$(LM3)$	$(\alpha.P) \lfloor\!\lfloor_{S,M} Q = \alpha.(P \parallel_{S,M} Q)$		$\alpha \notin S$
$(LM4)$	$(<f_i^+, w>.P) \lfloor\!\lfloor_{S,M} Q = <f_{n(M_f)}^+, w>.(P \parallel_{S,M \cup \{f:(n(M_f),l_i)\}} Q)$		
$(LM5)$	$(f_i^-.P) \lfloor\!\lfloor_{S,M} Q = f_j^-.(P \parallel_{S,M - \{f:(j,l_i)\}} Q)$		$f:(j, l_i) \in M$
$(LM6)$	$(P + Q) \lfloor\!\lfloor_{S,M} R = P \lfloor\!\lfloor_{S,M} R + Q \lfloor\!\lfloor_{S,M} R$		
$(SM1)$	$P \mid_{S,M} Q = Q \mid_{S,M} P$		
$(SM2)$	$\underline{0} \mid_{S,M} P = \underline{0}$		
$(SM3)$	$(\gamma.P) \mid_{S,M} (\gamma'.Q) = \underline{0}$		$(\gamma \notin S \ \vee \ \gamma \neq \gamma') \wedge \tau \notin \{\gamma, \gamma'\}$
$(SM4)$	$(\tau.P) \mid_{S,M} Q = P \mid_{S,M} Q$		
$(SM5)$	$(a.P) \mid_{S,M} (a.Q) = a.(P \parallel_{S,M} Q)$		$a \in S$
$(SM6)$	$(P + Q) \mid_{S,M} R = P \mid_{S,M} R + Q \mid_{S,M} R$		
$(Rec1)$	$recX.P = recY.(P\{Y/X\})$ provided that Y is not free in *rec*X.P		
$(Rec2)$	$recX.P = P\{recX.P/X\}$		
$(Rec3)$	$Q = P\{Q/X\} \Rightarrow Q = recX.P$ provided that X is strongly guarded in P		

Fig. 19. Axiomatization for IGSMP

A sequential state is defined to be one which includes "$\underline{0}$", "X" and operators ".", "+", "$recX$" only; leading to the following theorem.

Theorem 28. *If an IGSMP$_{sg}$ process P is finite state, then $\exists P' : \mathcal{A}_{IGSMP} \vdash P = P'$ with P' sequential state.*

For sequential states the axioms of $\mathcal{A}_{IGSMP}$ involved are just the standard axioms of [27], and the axioms for priority and probabilistic choice. From Theorem 28 we derive the completeness of $\mathcal{A}_{IGSMP}$.

Theorem 29. *$\mathcal{A}_{IGSMP}$ is complete for $\simeq$ over finite state IGSMP$_{sg}$ processes.*

Example 7. Let us consider the system $recX.f.X \parallel_{\emptyset} recX.f.X$ of the previous Example 6. In the following we show how this process can be turned into a sequential process. In the following we let f_i^+ stand for $<f_i^+,1>$ and we abbreviate $\mathcal{A}_{IGSMP} \vdash P = Q$ with $P = Q$. Moreover we let $P \equiv \mathit{rec}X.f.X$ and $P' \equiv f_1^-.\mathit{rec}X.f.X$. We note that $P = f.\mathit{rec}X.f.X = f_1^+.f_1^-.\mathit{rec}X.f.X$, by applying $(\mathit{Rec}2)$ and (TAct). In a first phase, we start with the initial state $P \,\|_{\emptyset,\emptyset}\, P$ and we express each state in terms of states it may reach after a single prefix. We have:

$$P \,\|_{\emptyset,\emptyset}\, P = f_1^+.(P' \,\|_{\emptyset,\{f:(1,l_1)\}}\, P) + f_1^+.(P' \,\|_{\emptyset,\{f:(1,l_1)\}}\, P)$$

by applying (Par), $(\mathit{LM}4)$ and $(\mathit{SM}3)$. From this equation we derive:

$$P \,\|_{\emptyset,\emptyset}\, P = <f_1^+,2>.(P' \,\|_{\emptyset,\{f:(1,l_1)\}}\, P)$$

by applying (Prob). Then, we have:

$$P' \,\|_{\emptyset,\{f:(1,l_1)\}}\, P = f_2^+.(P' \,\|_{\emptyset,\{f:(1,l_1),f:(2,r_1)\}}\, P')$$

by applying (Par), $(\mathit{LM}4)$, $(\mathit{LM}5)$, $(\mathit{SM}3)$ and $(\mathit{Pri}2)$. Then, we have:

$$P' \,\|_{\emptyset,\{f:(1,l_1),f:(2,r_1)\}}\, P' = f_1^-.(P \,\|_{\emptyset,\{f:(2,r_1)\}}\, P') + f_2^-.(P \,\|_{\emptyset,\{f:(1,r_1)\}}\, P')$$

by applying (Par), $(\mathit{LM}5)$ and $(\mathit{SM}3)$. From this equation we derive:

$$P' \,\|_{\emptyset,\{f:(1,l_1),f:(2,r_1)\}}\, P' = f_1^-.(P \,\|_{\emptyset,\{f:(2,r_1)\}}\, P') + f_2^-.(P' \,\|_{\emptyset,\{f:(1,l_1)\}}\, P)$$

by applying (Par), $(A1)$ and $(\mathit{SM}1)$ to $P \,\|_{\emptyset,\{f:(1,r_1)\}}\, P'$. Finally we have:

$$P \,\|_{\emptyset,\{f:(2,r_1)\}}\, P' = f_1^+.(P' \,\|_{\emptyset,\{f:(1,l_1),f:(2,r_1)\}}\, P')$$

by applying (Par), $(\mathit{LM}4)$, $(\mathit{LM}5)$, $(\mathit{SM}3)$ and $(\mathit{Pri}2)$. Now we perform a second phase where we generate recursive processes and we substitute states with equivalent terms. We start with $P \,\|_{\emptyset,\{f:(2,r_1)\}}\, P'$. Since the state itself does not occur in its equivalent term we do not have to generate any recursion. Substituting the state with its equivalent term in the other equations generates the new equation:

$$\begin{gathered} P' \,\|_{\emptyset,\{f:(1,l_1),f:(2,r_1)\}}\, P' = \\ f_1^-.f_1^+.(P' \,\|_{\emptyset,\{f:(1,l_1),f:(2,r_1)\}}\, P') + f_2^-.(P' \,\|_{\emptyset,\{f:(1,l_1)\}}\, P). \end{gathered}$$

Then we consider the state $P' \,\|_{\emptyset,\{f:(1,l_1),f:(2,r_1)\}}\, P'$. Since the state itself indeed occurs in its equivalent term, we have to change its equation by generating a recursion as follows:

$$P' \,\|_{\emptyset,\{f:(1,l_1),f:(2,r_1)\}}\, P' = \mathit{rec}Y.(f_1^-.f_1^+.Y + f_2^-.(P' \,\|_{\emptyset,\{f:(1,l_1)\}}\, P)).$$

Substituting the state with its equivalent term in the remaining equations generates the new equation:

$$P' \,\|_{\emptyset,\{f:(1,l_1)\}}\, P = f_2^+.\mathit{rec}Y.(f_1^-.f_1^+.Y + f_2^-.(P' \,\|_{\emptyset,\{f:(1,l_1)\}}\, P)).$$

Now we consider the state $P' \,\|_{\emptyset,\{f:(1,l_1)\}}\, P$. We change its equation by generating a recursion as follows:

$$P' \,\|_{\emptyset,\{f:(1,l_1)\}}\, P = \mathit{rec}X.(f_2^+.\mathit{rec}Y(f_1^-.f_1^+.Y + f_2^-.X)).$$

Substituting the state with its equivalent term in the remaining equations generates the new equation:

$$P \,\|_{\emptyset,\emptyset}\, P = <f_1^+,2>.\mathit{rec}X.(f_2^+.\mathit{rec}Y(f_1^-.f_1^+.Y + f_2^-.X)).$$

Therefore we have turned our initial system $recX.f.X \parallel_\emptyset recX.f.X$ into the recursive sequential process $<f_1^+, 2>.recX.(f_2^+.recY(f_1^-.f_1^+.Y + f_2^-.X))$. Note that the operational semantics of this process generates the labeled transition system of Fig. 10 derived in Sect. 2 as the minimal version of the IGSMP in Fig. 18. ■

4 Example: Queuing Systems G/G/1/q

In this section we present an example of specification with *IGSMP*. In particular we concentrate on Queuing Systems (*QSs*) $G/G/1/q$, i.e. *QSs* which have one server and a FIFO queue with q-1 seats and serve a population of unboundedly many customers. In particular the *QS* has an interarrival time which is generally distributed with distribution f and a service time which is generally distributed with distribution g.

Such a system can be modeled with the calculus of *IGSMPs* as follows. Let a be the action representing the fact that a new customer arrives at the queue of the service center, d be the action representing that a customer is delivered by the queue to the server. The process algebra specification is the following one: [13]

– $QS_{G/G/1/q} \triangleq Arrivals \parallel_{\{a\}} (Queue_0 \parallel_{\{d\}} Server)$

  - $Arrivals \triangleq f.a.Arrivals$
  - $Queue_0 \triangleq a.Queue_1$

    $Queue_h \triangleq a.Queue_{h+1} + d.Queue_{h-1} \qquad 0 < h < q-1$

    $Queue_{q-1} \triangleq a.Queue_{q-1} + d.Queue_{q-2}$
  - $Server \triangleq d.g.Server$

We have specified the whole system as the composition of the arrival process, the queue and the server which communicate via action types a and d. Then we have separately modeled the arrival process, the queue, and the server. As a consequence if we want to modify the description by changing the interarrival time distribution f or the service time distribution g , only component *Arrivals* or *Server* needs to be modified while component *Queue* is not affected. Note that the role of actions a and d is defining interactions among the different system components. Such actions have zero duration and they are neglected from the performance viewpoint.

In Fig. 20 we show $\mathcal{G}[\![QS_{G/G/1/q}]\!]$. In this picture A stands for *Arrivals*, A' stands for $f^-.a.Arrivals$, A'' stands for $a.Arrivals$. Similarly, S stands for *Server*, S' stands for $g.Server$, S'' stands for $g^-.Server$. Moreover, Q_h stands for $Queue_h$, for any h. We omit parallel composition operators in terms, so, e.g., AQ_hS stands for $Arrivals \parallel_{\{a\}} (Queue_h \parallel_{\{d\}} Server)$.

[13] In the specification we use process constants instead of the operator "*recX*", to denote recursion. The reason being that the use of constants is suitable for doing specifications, while the use of operator "*recX*" is preferable when dealing with axiomatizations. The two constructs are shown to be completely equivalent in [27].

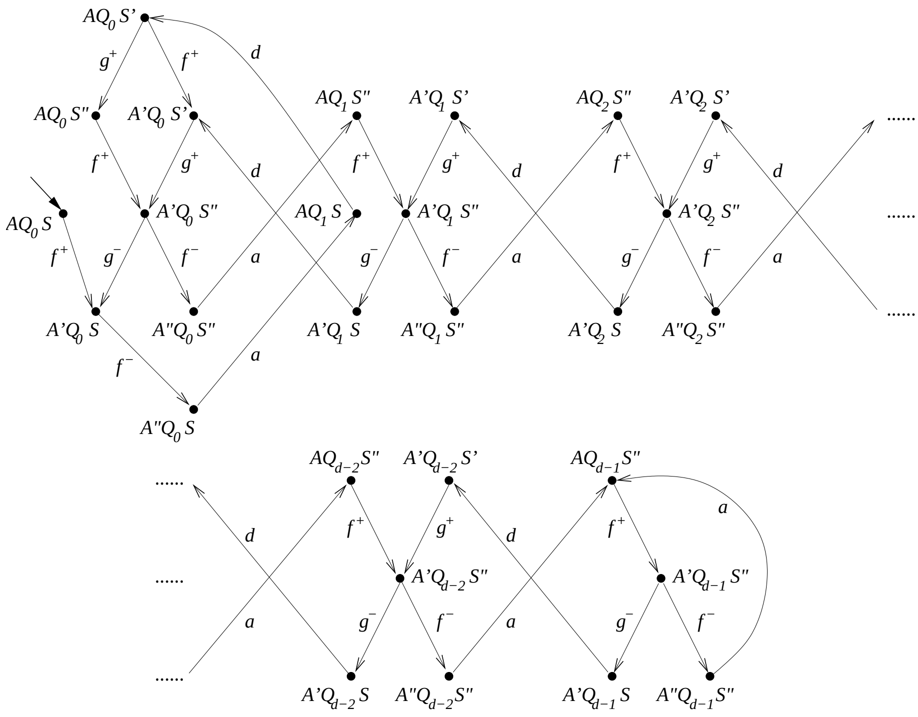


Fig. 20. Semantic Model

In order to derive the performance model of the system $QS_{G/G/1/q}$ we have to make sure that it is complete both from the interaction and the performance viewpoints. In Fig. 20 we have visible actions a and d, therefore the behavior of the system can be influenced by interaction with the environment and is not complete. We make it complete by considering $QS_{G/G/1/q}/\{a, d\}$ so that every action in the semantic model of Fig. 20 becomes a τ action. As far as completeness w.r.t. performance is concerned, we present in Fig. 21 the minimal version of $\mathcal{G}[\![QS_{G/G/1/q}/\{a, d\}]\!]$, obtained by aggregating weakly bisimilar states (see Sect. 3.3). Since in the minimal model there are no longer internal τ actions, we have that our system is complete also w.r.t. performance.

By applying the procedure described in Sect. 2.3, hence by solving choice trees in the minimal model of Fig. 21, we finally obtain the *GSMP* of Fig. 22. The elements e_1 and e_2 represent the delays f and g.

5 Conclusion

In this paper we have presented the calculus and the model of Interactive Generalised Semi-Markov Processes. Such a specification language, together with its associated weak bisimulation based theory, constitutes a solid formal basis for

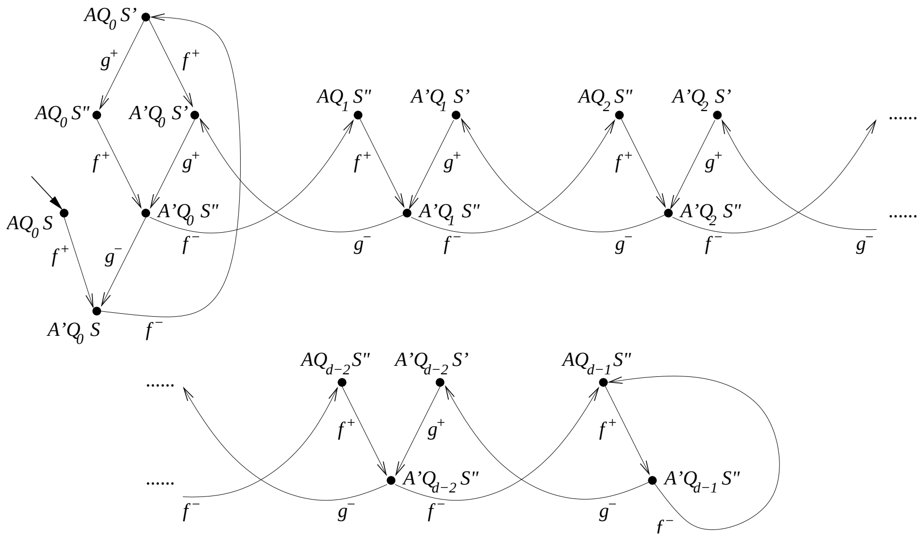
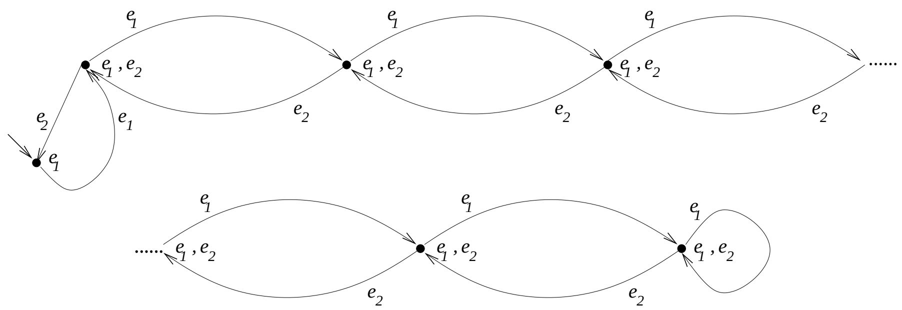


Fig. 21. Minimal Semantic Model



$$ElPDF(e_1) = f, \qquad ElPDF(e_2) = g$$

Fig. 22. Derived GSMP

specifying and analyzing concurrent/distributed systems with probabilistic generally distributed time, i.e. systems with real-time and stochastic time aspects. As far as system analysis is concerned, we showed that it is possible to derive from complete IGSMP specifications the underlying stochastic process in the form of a GSMP which can be analyzed with standard simulative or analytical techniques. In [6, 5, 8] we also show that from an IGSMP specification it is possible to derive: (i) a pure real-time specification in the form of a timed automata (by considering distribution supports only, thus loosing stochastic quantification over possible time values); and (ii) a pure Markovian specification in the form of a term of a Markovian process algebra (by approximating general distributions with phase-type distributions, thus loosing real-time constraints over time values).

As far as related work is concerned, several algebraic languages which express generally distributed durations like the calculus of *IGSMPs* have been developed. The languages that are closest to *IGSMP*, in that they produce semantic models which represent probabilistic durations via clocks/elements as in *GSMPs* are those of [2, 18]. Conceptually, they somehow correspond to different techniques for dealing with ST semantics. With the language of [2], performance models are derived from terms specifying systems by applying to them a preliminary procedure that gives a different name to each durational action of the term. In this way, each name represents a different clock in the semantic model of the system. In the approach of [2] the events of action starts are not explicitly expressed in the semantic models and choices are resolved via the race policy (alternative actions are executed in parallel and the first action that terminates wins) instead of the preselection policy as in *IGSMP*. The approach of [18], which has been developed concurrently with our work on general distributions (the first papers [19] and [11] on the two approaches appeared at the same workshop), is the most similar to ours. The language presented in [18] is endowed with an abstract semantics which may generate finite intermediate semantic models. With this language clock names must be explicitly expressed in the term that specify the system and the fact that a different name is used for each clock is ensured by alpha-converting clock names while deriving the semantic model similarly as with ST semantics (but no static/dynamic rule is defined for generating clock names). As in *IGSMP* the execution of a clock is represented by the events of clock start and clock termination, but here these two events must be explicitly expressed in the term specifying a system and they are not automatically generated by the operational semantics. Moreover, in the language of [18] choices among clock executions (pairs of start and termination events) are resolved via the race policy instead of the preselection policy as in *IGSMP* and probabilistic choices (which are a basic ingredient of *GSMPs*) cannot be expressed. A drawback of the approaches of [2, 18] w.r.t. the calculus of *IGSMP* is that there is no easy way to express equivalence of systems, hence to produce congruence results and axiomatizations. This is because in order to establish the equivalence of two systems it is necessary to associate in some way the names of the clocks used by one system with the names of the corresponding clocks used by the other one (the same phenomenon that happens when ST semantics is expressed via a static technique). In *IGSMP*, instead, names of clocks are dynamically generated by the operational semantics with a fixed rule. In this way equivalent systems get the same names for clocks and there is no need to associate names of clocks for establishing equivalence. We can, therefore, rely on standard (probabilistic) bisimulation and we have the opportunity to reuse existing results and tools.

References

1. L. Aceto, *"On "Axiomatising Finite Concurrent Processes" "* in SIAM Journal on Computing 23(4):852-863, 1994

2. M. Ajmone Marsan, A. Bianco, L. Ciminiera, R. Sisto, A. Valenzano, *"A LOTOS Extension for the Performance Analysis of Distributed Systems"*, in IEEE/ACM Trans. on Networking 2:151-164, 1994
3. R. Alur, C. Courcoubetis, D. Dill *"Model-Checking in Dense Real-Time"*, in Information and Computation 104:2-34, 1993
4. M. Bernardo, *"Theory and Application of Extended Markovian Process Algebra"*, Ph.D. Thesis, University of Bologna (Italy), 1999
5. M. Bravetti, *"Specification and Analysis of Stochastic Real-Time Systems"*, Ph.D. Thesis, University of Bologna (Italy), 2002. Available at `http://www.cs.unibo.it/~bravetti`
6. M. Bravetti, *"Towards the Integration of Real-Time and Probabilistic-Time Process Algebras"*, in Proc. of the *3rd European Research Seminar on Advances in Distributed Systems (ERSADS '99)*, Madeira Island (Portugal), April 1999
7. M. Bravetti, *"Revisiting Interactive Markov Chains"*, in Proc. of the *3rd Int. Workshop on Models for Time-Critical Systems (MTCS 2002)*, ENTCS 68(5), Brno (Czech Republic), August 2002
8. M. Bravetti, *"An Integrated Approach for the Specification and Analysis of Stochastic Real-Time Systems"*, in Proc. of the *3rd Int. Workshop on Models for Time-Critical Systems (MTCS 2002)*, ENTCS 68(5), Brno (Czech Republic), August 2002
9. M. Bravetti, A. Aldini, *"Non-Determinism in Probabilistic Timed Systems with General Distributions"*, in Proc. of the *2nd Int. Workshop on Models for Time-Critical Systems (MTCS 2001)*, ENTCS 52(3), Aalborg (Denmark), August 2001
10. M. Bravetti, M. Bernardo, *"Compositional Asymmetric Cooperations for Process Algebras with Probabilities, Priorities, and Time"*, in Proc. of the *1st Int. Workshop on Models for Time-Critical Systems (MTCS 2000)*, ENTCS 39(3), State College (PA), 2000
11. M. Bravetti, M. Bernardo, R. Gorrieri, *"From EMPA to GSMPA: Allowing for General Distributions"*, in Proc. of the *5th Int. Workshop on Process Algebras and Performance Modeling (PAPM '97)*, E. Brinksma and A. Nymeyer editors, pp. 17-33, Enschede (The Netherlands), June 1997
12. M. Bravetti, M. Bernardo, R. Gorrieri, *"Towards Performance Evaluation with General Distributions in Process Algebras"*, in Proc. of the *9th Int. Conf. on Concurrency Theory (CONCUR '98)*, D. Sangiorgi and R. de Simone editors, LNCS 1466:405-422, Nice (France), September 1998
13. M. Bravetti, R. Gorrieri, *"Deciding and Axiomatizing Weak ST Bisimulation for a Process Algebra with Recursion and Action Refinement"*, in ACM Transactions on Computational Logic 3(4): 465-520 (2002)
14. M. Bravetti, R. Gorrieri, *"The Theory of Interactive Generalized Semi-Markov Processes"*, in Theoretical Computer Science 282(1): 5-32 (2002)
15. N. Busi, R.J. van Glabbeek, R. Gorrieri, *"Axiomatising ST-Bisimulation Equivalence"*, in Proc. of the *IFIP Working Conf. on Programming Concepts, Methods and Calculi (PROCOMET '94)*, pp. 169-188, S. Miniato (Italy), 1994
16. C.G. Cassandras, *"Discrete Event Systems. Modeling and Performance Analysis"*, Aksen Associates, Irwin, 1993
17. D.R. Cox, *"The Analysis of non-Markovian Stochastic Processes by the Inclusion of Supplementary Variables"*, in Proc. of the Cambridge Philosophical Society 51:433-440, 1955
18. P.R. D'Argenio, *"Algebras and Automata for Timed and Stochastic Systems"*, Ph.D. Thesis, Univ. Twente, 1997

19. P.R. D'Argenio, J.-P. Katoen, E. Brinksma, *"A Stochastic Automata Model and its Algebraic Approach"* in Proc. of the *5th Workshop on Process Algebras and Performance Modelling (PAPM '97)*, pp. 1-16, Enschede (The Netherlands), 1997
20. R.J. van Glabbeek, F.W. Vaandrager, *"Petri Net Models for Algebraic Theories of Concurrency"*, in Proc. of the *Conf. on Parallel Architectures and Languages Europe (PARLE '87)*, LNCS 259:224-242, Eindhoven (The Netherlands), 1987
21. N. Götz, U. Herzog, M. Rettelbach, *"TIPP - A Stochastic Process Algebra"*, in Proc. of the *1st Workshop on Process Algebras and Performance Modelling (PAPM '93)*, pp. 31-36, Edinburgh (UK), 1993
22. H. Hermanns, *"Interactive Markov Chains"*, Ph.D. Thesis, Universität Erlangen-Nürnberg (Germany), 1998
23. J. Hillston, *"A Compositional Approach to Performance Modelling"*, Cambridge University Press, 1996
24. C.A.R. Hoare, *"Communicating Sequential Processes"*, Prentice Hall, 1985
25. K.G. Larsen, A. Skou, *"Bisimulation through Probabilistic Testing"*, in Information and Computation 94:1-28, 1991
26. K. Matthes, *"Zur Theorie der Bedienungsprozesse"*, in Trans. of the *3rd Prague Conf. on Information Theory, Stat. Dec. Fns. and Random Processes*, pp. 513-528, 1962
27. R. Milner, *"Communication and Concurrency"*, Prentice Hall, 1989
28. X. Nicollin, J. Sifakis, S. Yovine, *"Compiling Real-Time Specifications into Extended Automata"*, in IEEE Trans. on Software Engineering, 18(9):794-804, 1992
29. C. Priami, *"Stochastic π-Calculus with General Distributions"*, in *Proc. of the 4th Workshop on Process Algebras and Performance Modelling (PAPM '96)*, CLUT, pp. 41-57, Torino (Italy), 1996