Electronic Notes in Theoretical Computer Science 68 No. 5 (2003)
URL: http://www.elsevier.nl/locate/entcs/volume68.html 31 pages

An Integrated Approach for the Specification and Analysis of Stochastic Real-Time Systems

Mario Bravetti [1]

Dipartimento di Scienze dell'Informazione,
University of Bologna, Mura Anteo Zamboni 7, 40127 Bologna, Italy

Abstract

A formal approach for the specification and analysis of concurrent systems is proposed which integrates two different orthogonal aspects of time: (*i*) real-time, concerning the expression of time constraints and the verification of exact time properties, and (*ii*) probabilistic-time, concerning the probabilistic quantification of durations of system activities via exponential probability distributions and the evaluation of system performance. We show that these two aspects, that led to different specification paradigms called timed automata and Markovian process algebras, respectively, can be expressed in an integrated way by a single language: a process algebra capable of expressing activities with generally distributed durations. In particular, we consider the calculus of Interactive Generalized Semi-Markov Processes (IGSMPs) and we present formal techniques for compositionally deriving, from an IGSMP specification, (*i*) a pure real-time model (called Interactive Timed Automaton), by considering the support of general distributions, and (*ii*) a pure probabilistic-time model (called Interactive Weighted Markov Chain), by approximating general distributions with phase-type distributions.

1 Introduction

The importance of considering the behavior of concurrent systems with respect to time during their design process has been widely recognized [17,3,9,2,20,21]. 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. [17,3,15]). According to this approach the time 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

[1] Email: bravetti@cs.unibo.it

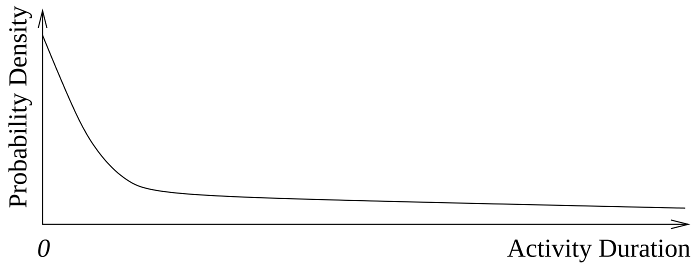

Fig. 1. Stochastic Time (Markovian) Activity

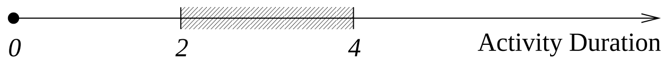

Fig. 2. Real-Time Activity

mathematical or simulative techniques. This approach has led to the definition of *stochastic process algebras*, an extension of standard process algebras [19] (concurrent specification languages which allow us to represent concurrent systems compositionally by specifying the behavior of individual processes and the way they interact) where a distribution of duration is associated with each action of a process. In most cases, as in [3], 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 [2,20,21] 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* [2]. 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. [2,20,21]).

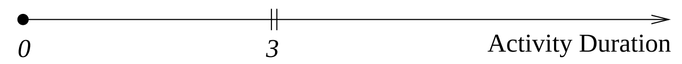

Fig. 3. Activity with a Fixed Duration

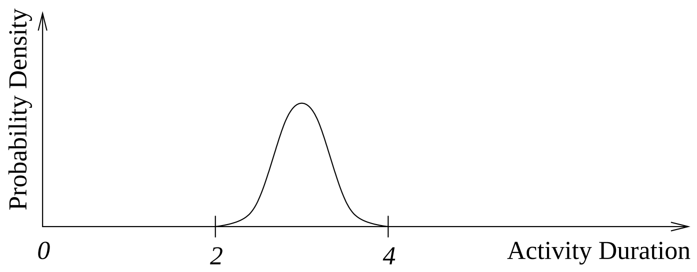

Fig. 4. Generally Distributed Activity

1.1 The Basic Idea

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, that we have introduced in [6], was subsequently applied also in [10] and [12]. 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

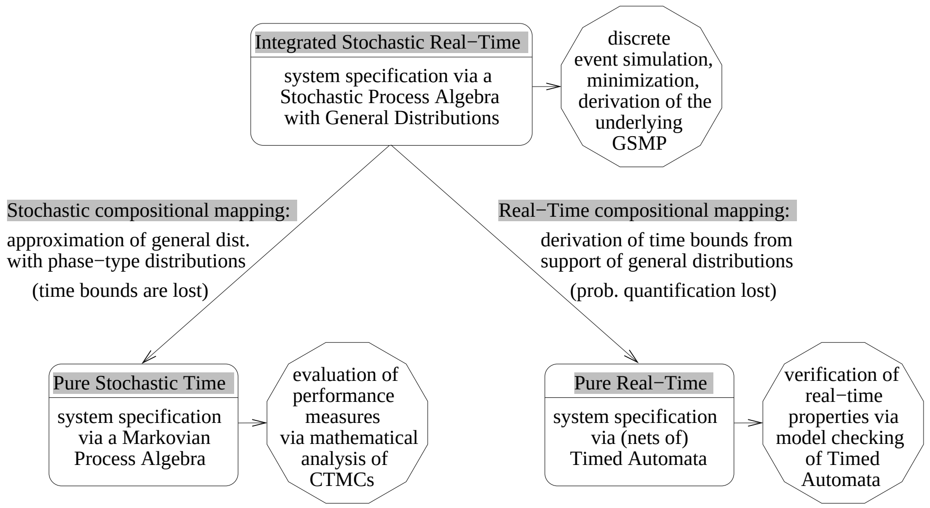

Fig. 5. Stochastic Real-Time Integrated Approach

single value of time.

1.2 An Integrated Approach

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).

The price to pay by using general distributions is the complexity of the stochastic process representing the system behavior: a *Generalized Semi-Markov Process* (GSMP). Only for very restricted cases we can derive performance measures from a GSMP by means of exact mathematical analysis.

As a consequence it is important that, besides developing a new stochastic real-time specification language by using generally distributed time and some new (usually complex and limited in power) analysis methodologies for such a language, we also develop formal automatizable procedures for deriving, from an integrated stochastic real-time specification, a traditional pure stochastic-time specification and a traditional pure real-time specification.

More in the details, in Fig. 5 we show how process algebra with generally distributed time can offer the possibility of such an integrated approach for the modeling and analysis of Stochastic Real-Time concurrent/distributed systems. Specifications (terms of such a process algebra) can be directly analyzed through standard discrete event simulation (see e.g. [13]), state space minimization (via a e.g. a notion of bisimulation based congruence), and derivation of the underlying performance model in the form of a GSMP. Besides the possibility of performing direct analysis, we can have formal techniques for compositionally deriving, from a system specification:

- A pure stochastic-time (Markovian) specification in the form of a term of a Markovian process algebra, by approximating general distributions with combinations of exponential distributions (the so called *phase-type* distributions). A consequence of this transformation is that all duration values for delays get probability greater than 0. Hence the information about time constraints (related to the real-time behavior of the system) is lost.
- A pure real-time specification in the form of a net (a parallel composition) of Timed Automata, by considering the support of general distributions, i.e. the set of time values that are given probability (density) greater than 0, and by turning probabilistic choices into non-deterministic choices. As a consequence the information related to the probabilistic-time behavior of the system is lost.

In this way whenever a user is interested in evaluating system properties which are related to the stochastic-time or real-time aspect only of the specified system, the analysis can be done automatically by deriving the specific traditional pure (stochastic-time or real-time) model and by analyzing it. This is very important from a practical viewpoint in that it gives the opportunity of reusing existing techniques and tools already developed for performance evaluation and model-checking of non-probabilistic real-time properties. Moreover, the advantage of deriving a traditional pure stochastic-time and real-time model from the same initial integrated specification (w.r.t. generating them independently) is that they are guaranteed to be consistent, in that they represent different aspects of the same initial system specification.

Example 1.1 Let us consider the following specification of a rail-road crossing. When a train is arriving at the railroad crossing, it sends a signal to the gate. When the gate receives the signal, it immediately starts emitting the stop signal to stop cars and (after a while) it closes. Then the gate waits for the train passage and opens up again. The time it takes for the train to arrive at the railroad crossing since it sends the approaching signal can be modeled by a probability distribution f (e.g. a distribution similar to a Gaussian distribution possibly with a lower and/or upper bounded support), the time it takes for the traffic to be stopped since the stop signal is displayed can be modeled by a probability distribution g (e.g. a distribution similar to an exponential distribution possibly with an upper bounded support) and, finally, the time between the passage of a train and the arrival of the next one can be modeled by a probability distribution h (e.g. a distribution similar to an exponential distribution). Given a formal algebraic specification of the railroad crossing, we could, e.g., evaluate it w.r.t. the following properties: we would like to be sure that it can never happen that the train traverses the railroad crossing before the traffic is stopped (safety real-time property) and we would like to evaluate the percentage of time in which the traffic is stopped (performance property). The former property can be evaluated by turning the initial stochastic real-time specification into a net of Timed Automata where only distribution supports are taken into account: it is easy to see that the unsafe "crash state" (where the train is traversing the railroad crossing while the traffic is not yet stopped) is reachable only if the upper bound of the support of distribution g is greater than the lower bound of the support of distribution f. On the other hand, the latter property can be evaluated by turning the initial specification into a Markovian algebraic specification where general distributions are approximated by phase types (e.g. we could use an Erlang distribution with 4,8 or more phases to approximate f, and exponential distributions to approximate g and h): the percentage of time in which the traffic is stopped is obtained by summing up the steady state probability of all the states of the resulting Markov chain in which the system is waiting for the delay f to expire but is not waiting for the delay g to expire (because it already expired). The more phases we use in the approximation, the more precise gets the percentage we get.

1.3 Implementing the Integrated Approach by means of Interactive Systems

In this paper we implement the integrated approach of Fig. 5 by using, as a process algebra with generally distributed time, the calculus of *Interactive Generalized Semi-Markov Processes* (IGSMPs) introduced in [9,7,5]. When considered at the level of transition systems, IGSMPs are basically an extension of GSMPs with *action transitions*, representing the ability of the process to interact with other processes. A concurrent/distributed system is, therefore, specified by a process algebraic term of the calculus of IGSMP by means of basic mechanisms like: parallel composition, internal probabilistic choices (not influenced by the environment), probabilistic time delays with a general distribution, non-determinism and interaction events (represented by actions).

Besides using discrete event simulation, systems specifications (terms of the calculus of IGSMPs) can be directly analyzed by means of the techniques introduced in [9]: minimization via a notion of bisimulation based congruence which abstracts from internal system activities (τ actions), and derivation of the underlying performance model in the form of a GSMP for IGSMPs which are complete both from the interactive and from the performance viewpoints. As far as the stochastic-time and real-time projections of Fig. 5 are concerned, we use the following specification paradigms:

- A pure stochastic-time (Markovian) specification is a term of the calculus of *Interactive Weighted Markov Chains* (IWMCs). Interactive Weighted Markov Chains are basically an extension of Continuous Time Markov Chains with *action transitions*, representing the ability of the process to interact with other processes, and *probabilistic transitions*, representing probabilistic choices internally performed by the process. In particular Interactive Weighted Markov Chains extend Interactive Markov Chains of [15] with the capability of representing probabilistic choices through transitions labeled with *weights* [23].

- A pure real-time specification is a net of *Interactive Timed Automata* (ITA). Interactive Timed Automata are a variant of classical Timed Automata [2,21], where action executions, events enabled on the basis of clock constraints and clock reset events are expressed by means of separate transitions. The advantage of ITA with respect to existing Timed Automata is that action transitions can be dealt with separately from time-related transitions, hence making it easy to define, e.g., a notion of weak bisimulation.

The technique leading to the derivation of the IWMC is particularly significant in that: (*i*) it shows process algebra to provide exactly the machinery necessary for approximating GSMPs with CTMCs through phase-type distributions, and (*ii*) it confirms ST semantics to be the adequate semantics for generally distributed time (as claimed e.g. in [9,5]) in that approximation of activity durations with phase-type distributions is a form of action refinement [1,8]. From the practical viewpoint the approximation of general distributions with phase-type distributions will cause an approximation on the obtained performance measures. In particular the measures obtained will tend to the exact measures as the approximating phase-type durations tend to the exact duration distributions (by increasing the number of phases considered in the approximating phase-types). The problem of evaluating the error introduced in the measures depending on the level of approximation is a very difficult and known problem of statistics (see e.g. [4]) whose solution is somehow orthogonal to the results presented in this paper. Moreover note that, the better the approximation is, the greater the state space explosion caused by phase-type expansion is. Obviously this may become a problem if we want to reach certain levels of precision. Again solutions of this well-known problem are somehow orthogonal to the contents of this paper, e.g. we could adopt the technique introduced in [22] where the state-space is represented with Kronecker matrix expressions. On the other hand in spite of its inconveniences, for most systems with general distributions, approximation with phase-type is the only practical way to do performance analysis not based on simulation.

As far as the mapping into ITA is concerned, it just turns probability distributions into set of possible values for clocks by using distribution supports (defined by adopting the technical shrewdness introduced in [12]) without modifying the "structure" of the transition system. Therefore it has the desirable property of not increasing the number of states of the IGSMP when translating it into an ITA. Such a simple technique, which cannot be correctly applied to the Stochastic Automata model of [11] (see [12]), is convenient w.r.t. the more complex one introduced in [12] in that it avoids a blow up in the number of states which is exponential in the number of clocks used in the initial specification (see Sect. 4.2 for the details). As discussed in more details in [12], the mappings based on supports like this one guarantee that each behavior of the IGSMP which was executable with probability greater than zero becomes a possible behavior of the resulting ITA, but in general the converse cannot be stated. Hence at least non-probabilistic real-time safety properties of the IGSMP can be checked in the resulting ITA. As far as liveness properties are concerned only some kind of them (e.g. those related to possible action behaviors and not to particular time values) can be shown to hold in the initial IGSMP.

Unfortunately, in order not to make presentation too long, we do not include in this paper the definition of IGSMPs and their semantics, which can be found in [7,5]. The same holds for the calculus of IGSMPs, which is simply a variant of the calculus of IMC [15] where prefixes $<f, w>$ (representing delays whose duration has general distribution f and are chosen according to weight w) are used instead of λ prefixes (representing exponentially timed delays of rate λ), and its semantics (which maps algebraic terms into IGSMPs) which are defined in [9,5].

1.4 Outline of the Paper

The paper is organized as follows. In Sect. 2 we introduce the calculus of IWMCs, which constitutes the first extension of IMC [15] with probabilistic choices endowed with a complete axiomatization for weak bisimulation. Then, in Sect. 3 we introduce ITA, which constitute the first variant of timed automata [2,21] endowed with a weak version of (structural) bisimulation equivalence and a compositional semantics. Finally, in Sect. 4 we present the two formal mappings from IGSMP specifications to IWMC and ITA specifications and we show that: (*i*) the IGSMP - IWMC mapping preserves performance measures once we replace generally distributed durations with the approximating phase-type durations in the initial IGSMP, (*ii*) the IGSMP - ITA mapping is such that the traces of "supported" behaviors (originating from time values in the support of distributions) starting in a state of the IGSMP are the same as the traces of possible behaviors starting in the corresponding state of the ITA (as in [12]), and (*iii*) both mappings are compositional and preserve weak bisimulation equivalence. In Appendix A we show an axiomatization for weak bisimulation which is complete over finite state IWMC terms, while in Appendix B we present the semantics of ITA and we show that it is compositional and preserves equivalence.

Proofs of theorems can be found in [5] Chapters 4,5 and 8.

2 Interactive Weighted Markov Chains

Interactive Weighted Markov Chains are an extension of Continuous Time Markov Chains with *action transitions*, representing the ability of the process to interact with other processes, and *probabilistic transitions*, representing probabilistic choices internally performed by the process. Action transitions, probabilistic transitions and exponential transitions of CTMCs give rise to different kind of states (where choices based on non-determinism, probability and time are performed) similarly as in an alternating model [24,14].

More precisely, Interactive Weighted Markov Chains extend Interactive Markov Chains of [15] with the capability of representing probabilistic choices through probabilistic transitions labeled with *weights* [23]. Extending IMCs in this way is very convenient in that it significantly simplifies the task of modeling real systems (in that alternative system behaviors can be expressed via probabilistic choices) without increasing the "complexity" of the underlying class of stochastic processes. This because probabilistic choices just give rise to vanishing states which can be eliminated via a simple procedure (see [3] Chapter 4) when evaluating performance.

Similarly to [15], in IWMCs the interrelation between standard action transitions and performance related transitions (probabilistic and exponentially timed transitions) is governed by the so-called *maximal progress assumption* [20]: the possibility of executing τ transitions prevents the execution performance related transitions, thus expressing that the system cannot wait if it has something internal to do. But differently from [15], where such a priority is captured in the definition of equivalence among IMCs, we prefer to express priority by cutting transitions which cannot be performed when defining and composing IWMCs (a solution also hinted in [16]). This allows us to obtain smaller system models and to define a notion of bisimulation among IWMCs more simply, without having to discard any transitions when establishing equivalence.

As for IMCs [15], we will compose in parallel several IWMCs via a CSP-like synchronization policy. Alternative τ transitions in an IWMC represent internal non-deterministic choices which are performed in zero time and can never be "resolved" through synchronization with other system components. On the contrary, visible actions a in an IWMC are seen as incomplete actions which wait for a synchronization with other system components (they represent potential interaction with the environment). Therefore the choice of such actions in any IWMC state is governed by an external form of non-determinism, as their execution completely depends on the environment. We will also make use of an hiding operator which turns (incomplete) visible action transitions of an IWMC into (complete) τ transitions.

2.1 Definition of Interactive Weighted Markov Chain

In an IWMC 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 IWMC 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.
- *probabilistic states*, enabling probabilistic transitions and (possibly) visible action transitions a only. In such states (also called vanishing states) the

IWMC just performs a probabilistic choice among the probabilistic transitions in zero time (proportionally to the weights labeling the transitions) and may potentially interact with the environment through one of the visible actions.

- *timed states*, enabling exponentially timed transitions and (possibly) visible action transitions a only. The IWMC sojourns in these states (also called tangible states) until one of the exponential delays terminates and the corresponding transition is performed. While the IWMC 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 IWMC remains indefinitely. It may, at any time, potentially interact with the environment through one of the outgoing visible action transitions.

In the following we present the formal definition of Interactive Weighted Markovian Transition System (IWMTS), then we will define interactive weighted Markov chains as IWMTSs possessing an initial state. Formally, rates, belonging to $\mathbb{R}^+$, are ranged over by $\lambda, \lambda', \ldots$ while weights, belonging to $\mathbb{R}^+$, are ranged over by $w, w', \ldots$. We use $\theta, \theta', \ldots$ to range over both rates and weights. Moreover, we denote the set of standard action types used in a IWMTS 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 states of an IWMTS is denoted by Σ, ranged over by $s, s', \ldots$. We assume the following abbreviations that will make the definition of IWMTSs easier. Let us suppose that $T \subseteq (\Sigma \times \mathit{Labels} \times \Sigma)$ is a transition relation, where Labels is a set of transition labels, ranged over by l. In the remainder 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 $s \not\xrightarrow{\;l\;}$ to stand for $\nexists s' : s \xrightarrow{\;l\;} s'$.

Definition 2.1 An Interactive Weighted Markovian Transition System (IWMTS) is a tuple $\mathcal{M} = (\Sigma, Act, T_w, T_e, T_a)$ with

- Σ a set of states,
- Act a set of standard actions,
- $T_w \subseteq (\Sigma \times \mathbb{R}^+ \times \Sigma)$, $T_e \subseteq (\Sigma \times \mathbb{R}^+ \times \Sigma)$, and $T_a \subseteq (\Sigma \times Act \times \Sigma)$ three transition relations, containing probabilistic, exponentially timed and action transitions, respectively, such that: [2]

(i) $\forall s \in \Sigma.\ s \xrightarrow{\;\tau\;} \implies \nexists \theta.\, s \xrightarrow{\;\theta\;}$

(ii) $\forall s \in \Sigma.\ \exists w.\, s \xrightarrow{\;w\;} \implies \nexists \lambda.\, s \xrightarrow{\;\lambda\;}$

[2] 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.

An Interactive Weighted Markov Chain (IWMC) is a tuple $\mathcal{M} = (\Sigma, Act, T_w, T_e, T_a, s_0)$, where $s_0 \in \Sigma$ is the initial state of the IWMC and $(\Sigma, Act, T_w, T_e, T_a)$ is an IWMTS. ■

The constraints over transition relations T_w, T_e and T_a guarantee that each state of the IWMC 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 exponentially timed or probabilistic transitions. 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 probabilistic transitions then it cannot perform exponentially timed transitions. Such a property derives from the assumption of *urgency of choices*: probabilistic choices cannot be delayed but must be performed immediately, hence they prevent the execution of exponentially timed delays.

2.2 The Calculus of IWMCs

Let Var be a set of process variables ranged over by X, Y, Z. Let $ARFun = \{\varphi : Act \longrightarrow Act \mid \varphi(\tau) = \tau \wedge \varphi(Act - \{\tau\}) \subseteq Act - \{\tau\}\}$ be a set of *action relabeling functions*, ranged over by φ.

Definition 2.2 We define the language $IWMC$ as the set of terms generated by the following syntax

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

where $L, S \subseteq Act - \{\tau\}$. An $IWMC$ process is a closed term of $IWMC$. We denote by $IWMC_g$ the set of strongly guarded terms of $IWMC$. [3] ■

"$\underline{0}$" denotes a process that cannot move. The operators "." and "+" are the CCS prefix and 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.

The semantics of IWMC terms produces a transition system labeled by actions in Act, weights in $\mathbb{R}^+$ and rates in $\mathbb{R}^+$. We use $\gamma, \gamma', \ldots$ to range over transition labels. Such a transition system is defined as being the $IWMTS$ $\mathcal{M} = (IWMC_g, Act, T_w, T_e, T_a)$, where: T_a is the least subset of $IWMC_g \times Act \times IWMC_g$ satisfying the standard operational rules of Table 1, T_w is obtained from the least multiset over $IWMC_g \times \mathbb{R}^+ \times IWMC_g$ satisfying the operational rules of Table 2 (similarly to [17,15], 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_e is obtained from the least multiset over $IWMC_g \times \mathbb{R}^+ \times IWMC_g$ satisfying the operational rules of Table 3 by summing the rates of the multiple occurrences

[3] We consider w and λ prefixes as being guards in the definition of strong guardedness.

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

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

$$\frac{P \xrightarrow{\alpha} P'}{P \parallel_S Q \xrightarrow{\alpha} P' \parallel_S Q}\ \alpha \notin S \qquad \frac{Q \xrightarrow{\alpha} Q'}{P \parallel_S Q \xrightarrow{\alpha} P \parallel_S Q'}\ \alpha \notin S$$

$$\frac{P \xrightarrow{a} P' \quad Q \xrightarrow{a} Q'}{P \parallel_S Q \xrightarrow{a} P' \parallel_S Q'}\ a \in S$$

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

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

Table 1
Standard Rules

$$w.P \xrightarrow{w} P$$

$$\frac{P \xrightarrow{w} P' \wedge Q \stackrel{\tau}{\not\longrightarrow}}{P + Q \xrightarrow{w} P'} \qquad \frac{Q \xrightarrow{w} Q' \wedge P \stackrel{\tau}{\not\longrightarrow}}{P + Q \xrightarrow{w} Q'}$$

$$\frac{P \xrightarrow{w} P' \wedge Q \stackrel{\tau}{\not\longrightarrow}}{P \parallel_S Q \xrightarrow{w} P' \parallel_S Q} \qquad \frac{Q \xrightarrow{w} Q' \wedge P \stackrel{\tau}{\not\longrightarrow}}{P \parallel_S Q \xrightarrow{w} P \parallel_S Q'}$$

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

$$\frac{P\{recX.P/X\} \xrightarrow{w} P'}{recX.P \xrightarrow{w} P'}$$

Table 2
Rules for Probabilistic Moves

of the same transition. In Tables 2 and 3 we use $P \xrightarrow{a}$ to stand for $\exists P' : P \xrightarrow{a} P'$, $P \stackrel{\tau}{\not\longrightarrow}$ to stand for $\nexists Q : P \xrightarrow{\tau} Q$ and $P \stackrel{w}{\not\longrightarrow}$ to stand for $\nexists w, Q : P \xrightarrow{w} Q$.

The rules of Table 2 define probabilistic transitions, by taking into account

$$\lambda.P \xrightarrow{\lambda} P$$

$$\frac{P \xrightarrow{\lambda} P' \wedge Q \stackrel{\tau}{\not\longrightarrow} \wedge Q \stackrel{w}{\not\longrightarrow}}{P + Q \xrightarrow{\lambda} P'} \qquad \frac{Q \xrightarrow{\lambda} Q' \wedge P \stackrel{\tau}{\not\longrightarrow} \wedge P \stackrel{w}{\not\longrightarrow}}{P + Q \xrightarrow{\lambda} Q'}$$

$$\frac{P \xrightarrow{\lambda} P' \wedge Q \stackrel{\tau}{\not\longrightarrow} \wedge Q \stackrel{w}{\not\longrightarrow}}{P \parallel_S Q \xrightarrow{\lambda} P' \parallel_S Q} \qquad \frac{Q \xrightarrow{\lambda} Q' \wedge P \stackrel{\tau}{\not\longrightarrow} \wedge P \stackrel{w}{\not\longrightarrow}}{P \parallel_S Q \xrightarrow{\lambda} P \parallel_S Q'}$$

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

$$\frac{P\{recX.P/X\} \xrightarrow{\lambda} P'}{recX.P \xrightarrow{\lambda} P'}$$

Table 3
Rules for Exponentially Timed Moves

the priority of "τ" actions over weights. Note that we consider a "global" kind of weights which are applied also across the parallel operator. Moreover we can just interleave parallel weight transitions because they are executed in zero time.

Definition 2.3 The semantic model $\mathcal{M}[\![P]\!]$ of $P \in IWMC_g$ is the $IWMC$ defined by $\mathcal{M}[\![P]\!] = (\Sigma_P, Act, T_{w,P}, T_{e,P}, T_{a,P}, P)$, where: Σ_P is the least subset of $IWMC_g$ such that $P \in \Sigma_P$ and, if $P' \in \Sigma_P$ and $P' \xrightarrow{\gamma} P''$, then $P'' \in \Sigma_P$; moreover $T_{w,P}, T_{e,P}$ and $T_{a,P}$ are the restriction of T_w, T_e and T_a to $\Sigma_P \times Act \times \Sigma_P$, $\Sigma_P \times \mathbb{R}^+ \times \Sigma_P$ and $\Sigma_P \times \mathbb{R}^+ \times \Sigma_P$.

2.3 Observational Congruence for IWMCs

Observational congruence over IWMCs deals with exponentially timed choices according to Markovian bisimulation [17], deals with probabilistic choices according to probabilistic bisimulation [18], and abstracts from standard τ actions as in [19].

In our context we express cumulative probabilities and cumulative exponential times by aggregating weights and rates, respectively. In particular, if I is a set of states, $TW(s, I)$ represents the cumulative weight of probabilistic transitions leaving s and going into a state of I. Similarly, $TR(s, I)$ represents the cumulative rate of exponentially timed transitions from s to I.

The definition of weak bisimilarity is an adaptation of that presented in [15] to our context.

Let $\stackrel{\alpha}{\Longrightarrow}$ denote $(\stackrel{\tau}{\longrightarrow})^* \stackrel{\alpha}{\longrightarrow} (\stackrel{\tau}{\longrightarrow})^*$, i.e. a sequence of transitions including a single α transition and any number of τ transitions. Moreover we define $\stackrel{\hat{\alpha}}{\Longrightarrow} = \stackrel{\alpha}{\Longrightarrow}$ if $\alpha \neq \tau$ and $\stackrel{\hat{\tau}}{\Longrightarrow} = (\stackrel{\tau}{\longrightarrow})^*$, i.e. a possibly empty sequence of τ transitions.

Definition 2.4 Let $\mathcal{M} = (\Sigma, Act, T_w, T_e, T_a)$ be a IWMTS. An equivalence relation β on Σ is a *weak bisimulation* iff $s_1 \; \beta \; s_2$ implies

- for every $\alpha \in Act$ and $s_1' \in \Sigma$,
  $s_1 \stackrel{\alpha}{\longrightarrow} s_1'$ implies $s_2 \stackrel{\hat{\alpha}}{\Longrightarrow} s_2'$ for some s_2' with $s_1' \; \beta \; s_2'$,
- $TW(s_1, \Sigma) \neq \emptyset$ implies
  $s_2 \stackrel{\hat{\tau}}{\Longrightarrow} s_2'$ for some s_2' such that, for every equivalence class I of β,
  $TW(s_1, I) = TW(s_2', I)$
- $TR(s_1, \Sigma) \neq \emptyset$ implies
  $s_2 \stackrel{\hat{\tau}}{\Longrightarrow} s_2'$ for some s_2' such that, for every equivalence class I of β,
  $TR(s_1, I) = TR(s_2', I)$

Two states s_1 and s_2 are weakly bisimilar, denoted by $s_1 \approx_{IWMC} s_2$, iff (s_1, s_2) is included in some weak bisimulation. ■

Differently from [15], for the sake of simplicity, we do not express conditions about the stability of bisimilar states because we are interested in obtaining a congruence result only for strongly guarded processes of our calculus. Such processes cannot produce an IWMC which is forced in a τ loop, hence we do not have to recognize this situation.

Definition 2.5 Two closed terms P, Q of $IWMC_g$ are observational congruent, written $P \simeq_{IWMC} Q$, iff:

- for every $\alpha \in Act$ and $P' \in IWMC_g$,
  $P \stackrel{\alpha}{\longrightarrow} P'$ implies $Q \stackrel{\alpha}{\Longrightarrow} Q'$ for some Q' with $P' \approx_{IWMC} Q'$,
- for every $\alpha \in Act$ and $Q' \in IWMC_g$,
  $Q \stackrel{\alpha}{\longrightarrow} Q'$ implies $P \stackrel{\alpha}{\Longrightarrow} P'$ for some P' with $P' \approx_{IWMC} Q'$,
- for every equivalence class I of β,
  $TW(P, I) = TW(Q, I)$ and $TR(P, I) = TR(Q, I)$ ■

Theorem 2.6 *$\simeq_{IWMC}$ is a congruence over terms of $IWMC_g$ [4] w.r.t. all the operators of IWMC, including recursion.*

It is easy to to produce an axiomatization for $\simeq_{IWMC}$ which is complete over finite-state $IWMC_g$ terms (due to lack of space we refer to Appendix A for the details).

[4] Actually the congruence property holds for a wider class of processes which has the following semantical characterization: processes whose states may reach via (zero or more) "τ" transitions a state which cannot perform "τ" transitions.

3 Interactive Timed Automata

Interactive Timed Automata are a variant of classical Timed Automata [2,21], where action executions, events enabled on the basis of clock constraints and clock reset events are expressed by means of separate transitions (thus following the approach of [24,14,15]). The advantage of ITA with respect to existing timed automata, where usually we have one single kind of transition expressing all these features in a combined fashion, is that action transitions can be dealt with separately from time-related transitions, hence making it easy to define, e.g., a notion of weak bisimulation as a simple extension of the standard notion of [19]. Therefore, with respect to the existing equivalence notions for timed automata, abstracting from τ transitions, improves the capability of minimizing the state space of specified systems. ITA can be straightforwardly mapped into existing timed automata (e.g. those defined in [21]), hence previous decidability results and software tools can be exploited for analysing real-time properties in ITA specifications.

Time delays are modeled in ITA by means of clocks C_n which are set to zero and count upwards while time passes. An ITA represents the behavior of a system component by employing both clock reset transitions and clock bound transitions, representing the timed behavior of the component and standard action transitions, representing the interactive behavior of the component. Clock reset transitions are labeled with a clock name C_n and represent the event of reset of the clock (which is set to zero). After such event, C_n counts upwards while time passes and states are traversed by the automaton. When several clock reset transitions are enabled in an ITA state, the choice among them is just non-deterministic. Clock bound transitions are labeled with a clock constraint ϕ (an expression built from bounds on the clock values) and they can be executed only when the status of clocks satisfies such a constraint. A system is allowed to stay in a state enabling several clock bound transitions as long as *all* clock constraints labeling the transitions can be satisfied at present time or in the future. The role and the meaning of visible and invisible action transitions, related to composition of ITA via a CSP-like parallel composition and hiding, is exactly the same as for IWMCs.

3.1 Definition of Interactive Timed Automaton

In an ITA we have four different kinds of state:

- *silent states*, enabling invisible action transitions τ and (possibly) visible action transitions a only. The meaning of such states is exactly as in IWMCs.
- *reset states*, enabling reset transitions C_n and (possibly) visible action transitions a only. In such states the ITA just performs a choice among the clock reset transitions in zero time and may potentially interact with the environment through one of the visible actions.

- *timed states*, enabling clock bound transitions ϕ and (possibly) visible action transitions a only. In such states all the clocks of the ITA count upwards as time passes. The system is allowed to sojourn in the state as long as *all* clock constraints labeling its outgoing transitions can be satisfied at the present time or in the future. Moreover, it can non-deterministically leave the state at any time through a bound transition ϕ whose constraint ϕ is (at present time) satisfied. Moreover, while the ITA 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 ITA remains indefinitely. It may, at any time, potentially interact with the environment through one of the outgoing visible action transitions.

In the following we present the formal definition of Interactive Timed Automaton Transition System (ITATS), then we will define Interactive Timed Automata as ITATSs possessing an initial state. Formally, we use $T, T', \ldots$, representing sets of time values, to range over subsets of $\mathbb{R}^+ \cup \{0\}$. Moreover, we denote the set of standard action types used in an ITATS 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 ITATS 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 $\mathcal{C}^\Phi$, ranged over by $\phi, \phi', \ldots$, the set of constraints over clocks of $\mathcal{C}$ (the labels of clock bound transitions), which is defined as the set of terms generated by the following syntax:

$$\phi ::= C_n \in T \mid \phi \wedge \phi$$

Moreover, let $\mathcal{C} \cup \mathcal{C}^\Phi$ be ranged over by $\theta, \theta', \ldots$. The set of states of an ITATS is denoted by Σ, ranged over by $s, s', \ldots$. We assume the following abbreviations that will make the definition of ITATSs easier.

Definition 3.1 An Interactive Timed Automata Transition System (ITATS) is a tuple $\mathcal{T} = (\Sigma, \mathcal{C}, Act, T_r, T_b, T_a)$ with

- Σ a set of states,
- $\mathcal{C}$ a set of clocks,
- Act a set of standard actions,
- $T_r \subseteq (\Sigma \times \mathcal{C} \times \Sigma)$, $T_b \subseteq (\Sigma \times \mathcal{C}^\Phi \times \Sigma)$, and $T_a \subseteq (\Sigma \times Act \times \Sigma)$ three transition relations representing clock reset and clock bound events and action execution, respectively, such that:

(i) $\forall s \in \Sigma. \quad s \xrightarrow{\tau} \implies \nexists \theta.\, s \xrightarrow{\theta}$

(ii) $\forall s \in \Sigma. \quad \exists C_n.\, s \xrightarrow{C_n} \implies \nexists \phi.\, s \xrightarrow{\phi}$

An Interactive Timed Automata (ITA) is a tuple $\mathcal{T} = (\Sigma, \mathcal{C}, Act, T_r, T_b, T_a, s_0)$, where $s_0 \in \Sigma$ is the initial state of the ITA and $(\Sigma, \mathcal{C}, Act, T_r, T_b, T_a)$ is an ITATS. ■

The constraints over transition relations T_r, T_b and T_a guarantee that each state of the ITA 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 clock reset transitions or clock bound transitions. Such a property derives from the assumption of *maximal progress*: the possibility of performing internal actions prevents the execution of time-related activity. The second requirement says that if a state can perform clock reset transitions then it cannot perform clock bound transitions. Such a property derives from an assumption of *urgency of clock resets*: clock reset transitions cannot be delayed but must be performed immediately and they are just assumed to prevent the execution of clock bound transitions.

3.2 Composing ITA

In the following we present the formal definitions of parallel composition and hiding of ITA. It can be easily shown that the transition system obtained by the composition is still an ITA (see [5]) due to the fact that maximal progress and urgency of clock resets assumptions are enforced when composing ITA.

Given a clock renaming function $ren : \mathcal{C} \longrightarrow \mathcal{C}$, we assume $ren(\phi)$ to be the constraint ϕ' obtained from ϕ by renaming clocks in ϕ according to function ren. In particular we define the renaming function $l : \mathcal{C} \longrightarrow \mathcal{C}$ by $\{(C_n, C_{n,l}) \mid C_n \in \mathcal{C}\}$ and, similarly, function $r : \mathcal{C} \longrightarrow \mathcal{C}$ by $\{(C_n, C_{n,r}) \mid C_n \in \mathcal{C}\}$.

Definition 3.2 The parallel composition $\mathcal{T}_1 \|_S \mathcal{T}_2$ of two ITA $\mathcal{T}_1 = (\Sigma_1, \mathcal{C}_1, Act, T_{r,1}, T_{b,1}, T_{a,1}, s_{0,1})$ and $\mathcal{T}_2 = (\Sigma_2, \mathcal{C}_2, Act, T_{r,2}, T_{b,2}, T_{a,2}, s_{0,2})$, with $S \subset Act - \{\tau\}$ being the synchronization set, is the tuple $(\Sigma, \mathcal{C}, Act, T_r, T_b, T_a, (s_{0,1}, s_{0,2}))$ with

- $\Sigma = \Sigma_1 \times \Sigma_2 \times \mathcal{M}$ the set of states,
- $\mathcal{C} = \{C_{n,l} \mid C_n \in \mathcal{C}_1\} \cup \{C_{n,r} \mid C_n \in \mathcal{C}_2\}$
- $T_r \subseteq (\Sigma \times \mathcal{C} \times \Sigma)$, $T_b \subseteq (\Sigma \times \mathcal{C}^\Phi \times \Sigma)$, and $T_a \subseteq (\Sigma \times Act \times \Sigma)$ are the least transition relations, such that $\forall (s_1, s_2) \in \Sigma$.

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

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

$\mathbf{3_l}$ $s_1 \xrightarrow{C_n} s_1' \ \wedge\ s_2 \stackrel{\tau}{\not\longrightarrow} \implies (s_1, s_2) \xrightarrow{C_{n,l}} (s_1', s_2)$

$\mathbf{4_l}$ $s_1 \xrightarrow{\phi} s_1' \wedge s_2 \stackrel{\tau}{\not\longrightarrow} \wedge\ \not\exists C_n.\, s_2 \xrightarrow{C_n} \implies (s_1, s_2) \xrightarrow{l(\phi)} (s_1', s_2)$

  and also the symmetric rules $\mathbf{1_r}, \mathbf{3_r}, \mathbf{4_r}$ referring to the local transitions of $\mathcal{T}_2$, 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'), by turning l into r in the subscripts of clocks, and by turning the renaming function l into r, hold true.

- $(s_{0,1}, s_{0,2}) \in \Sigma$ the initial state ■

Each state $s \in \Sigma$ of the composed model is represented by a pair of states ($s_1 \in \Sigma_1$ and $s_2 \in \Sigma_2$). Moreover we rename clocks of both ITA $\mathcal{T}_1$ and $\mathcal{T}_2$

so to avoid a name conflict whenever two clocks with the same name C_n are simultaneously in execution in both ITA. 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) ITA, when $\alpha \notin S$ ($\alpha \in S$). Rules **3** and **4** define the behavior of the composed model in the case of clock reset and clock bound transitions, respectively, locally performed by components. Note that the negative clauses in the premises enforce the maximal progress and the urgency of clock resets assumptions.

Definition 3.3 The hiding $\mathcal{T}/L$ of a ITA $\mathcal{T} = (\Sigma, \mathcal{C}, Act, T_{r,1}, T_{b,1}, T_{a,1}, s_0)$ with $L \subset Act - \{\tau\}$ being the set of visible actions to be hidden is the tuple $(\Sigma, \mathcal{C}, Act, T_r, T_b, T_a, s_0)$ where $T_r \subseteq (\Sigma \times \mathcal{C} \times \Sigma)$, $T_b \subseteq (\Sigma \times \mathcal{C}^{\Phi} \times \Sigma)$, and $T_a \subseteq (\Sigma \times Act \times \Sigma)$ are the least set of transitions, such that $\forall s \in \Sigma$. [5]

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 $s \xrightarrow{\theta}_1 s' \wedge \ \nexists a \in L.\, s \xrightarrow{a}_1 \implies s \xrightarrow{\theta} s'$ ■

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

3.3 Weak bisimulation for ITA

Now we will introduce a notion of weak bisimulation over ITA which matches the clock related transitions as in [1] and abstracts from standard τ actions similarly to [19].

Given an ITATS $\mathcal{T} = (\Sigma, \mathcal{C}, Act, T_r, T_b, T_a)$, weak bisimulation over states is defined by associating clock names as in [1] so that equivalence does not depend on the particular names used for clocks. We use H to range over association histories of clock names, i.e. partial bijections from $\mathcal{C}$ to $\mathcal{C}$. We denote by $\mathcal{H}$ the set of all association histories.

We now present weak bisimulation for ITA which is defined by means of a family of bisimulations β_H, each indexed by an association history. First of all, let us say that a $\mathcal{H}$-indexed family of binary relations $\{\beta_H \mid H \in \mathcal{H}\}$ over Σ is *symmetric* if and only if $(s_1, s_2) \in \beta_H$ implies $(s_2, s_1) \in \beta_{\overline{H}}$, where $\overline{H} = \{(C_{n'}, C_n) \mid (C_n, C_{n'}) \in H\}$. Moreover, we use $H \leftarrow (C_n, C_{n'})$ to denote the association history H' obtained from H by adding the pair $(C_n, C_{n'})$ and removing old associations $(C_n, C_{n''})$ and $(C_{n'''}, C_{n'})$, for some $C_{n''}$ and $C_{n'''}$, already contained in H, thus preserving the structure of bijection from $\mathcal{C}$ to $\mathcal{C}$. We use γ to range over transition labels, i.e. $Act \cup \mathcal{C} \cup \mathcal{C}^{\Phi}$. Let $\stackrel{\gamma}{\Longrightarrow}$ denote $(\xrightarrow{\tau})^* \xrightarrow{\gamma} (\xrightarrow{\tau})^*$, i.e. a sequence of transitions including a single γ

[5] In order to distinguish transition of $T_{r,1}$, $T_{b,1}$ and $T_{a,1}$ from transitions of T_r, T_b and T_a we denote the former with "$\longrightarrow_1$" and the latter simply with "$\longrightarrow$".

transition and any number of τ transitions.

Definition 3.4 Let $\mathcal{T} = (\Sigma, \mathcal{C}, Act, T_r, T_b, T_a)$ be a ITATS. A symmetric $\mathcal{H}$-indexed family $\mathbf{B} = \{\beta_H \subseteq \Sigma \times \Sigma \mid H \in \mathcal{H}\}$ of binary relations over Σ is a *weak bisimulation family* iff $s_1 \; \beta_H \; s_2$ implies

- for every $\alpha \in Act$ and $s'_1 \in \Sigma$,
  $$s_1 \xrightarrow{\alpha} s'_1 \text{ implies } s_2 \stackrel{\hat{\alpha}}{\Longrightarrow} s'_2 \text{ for some } s'_2 \text{ with } s'_1 \; \beta_H \; s'_2$$
- for every $C_n \in \mathcal{C}$ and $s'_1 \in \Sigma$,
  $$s_1 \xrightarrow{C_n} s'_1 \text{ implies } s_2 \stackrel{C_{n'}}{\Longrightarrow} s'_2 \text{ for some } s'_2, C_{n'} \text{ with } s'_1 \; \beta_{H \leftarrow (C_n, C_{n'})} \; s'_2$$
- for every $\phi \in \mathcal{C}^{\Phi}$ and $s'_1 \in \Sigma$,
  $$s_1 \xrightarrow{\phi} s'_1 \text{ implies } \phi \in dom(H) \text{ and } s_2 \stackrel{H(\phi)}{\Longrightarrow} s'_2 \text{ for some } s'_2 \text{ with } s'_1 \; \beta_H \; s'_2$$

Two states s_1 and s_2 are weakly bisimilar with respect to association history $H \in \mathcal{H}$, denoted by $s_1 \approx_{ITA,H} s_2$, iff there exist some weak bisimulation family $\mathbf{B} = \{\beta_H \mid H \in \mathcal{H}\}$ such that $(s_1, s_2) \in \beta_H$. Two ITA $(\mathcal{T}_1, s_{0,1})$ and $(\mathcal{T}_2, s_{0,2})$ are weakly bisimilar, denoted by $(\mathcal{T}_1, s_{0,1}) \approx_{ITA} (\mathcal{T}_2, s_{0,2})$ if their initial states $s_{0,1}$ and $s_{0,2}$ are such that $s_{0,1} \approx_{ITA,\emptyset} s_{0,2}$ in the ITATS obtained with the disjoint union of $\mathcal{T}_1$ and $\mathcal{T}_2$. ■

Theorem 3.5 $\approx_{ITA}$ *is a congruence over ITA whose states may reach via (zero or more) "τ" transitions a state which cannot perform "τ" transitions w.r.t. both parallel and hiding.* ■

3.4 Semantics of ITA

ITA are endowed with a semantics which maps an ITA onto a transition system where: (i) the passage of time is explicitly represented by transitions labeled with numeric time delays $t \in \mathbb{R}^+ \cup 0$ and (ii) clock reset transitions and clock bound transitions are turned into prioritized transitions reflecting the precedence of clock reset transitions over clock bound transitions. Differently from existing approaches, we express semantic models of ITA by means of "interactive" 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 and preserves equivalence, similarly to what is done in [7,5] for IGSMPs. Due to lack of space we refer the reader to Appendix B for a complete presentation of the semantics of ITA.

4 Mapping IGSMPs onto Pure Markovian and Real-Time Processes

In this section we present the two formal mappings from IGSMPs, representing the stochastic and real-time behavior of a concurrent system in an integrated way, into IWMCs, representing the pure stochastic (Markovian) behavior of the system, and into ITA, representing the pure real-time behavior of the system. The former mapping is obtained by approximating generally distributed durations with phase-type durations. Technically, such mapping is performed compositionally at the algebraic level by replacing each delay prefix $<f, w>$ occurring in an algebraic term of an IGSMP specification with an IWMC term $w.P$, where P is the algebraic representation of a phase-type distribution approximating f. In this way we map a term of the calculus of IGSMPs into a term of IWMC. The latter mapping is obtained by abstracting from probability related information. Such mapping is still performed compositionally, but at the level of models (not at the level of algebraic terms). In particular we define how to derive an ITA from an IGSMP by turning probabilistic choices into non-deterministic choices and by considering the support of the distribution of a clock, i.e. the set of time values that may happen with probability (density) greater than 0, as the set of possible values for its duration. Moreover we show that such mapping is compositional, i.e. is preserved by CSP parallel composition and hiding. If every distribution used in the GSMP has a support which is a finite collection of intervals, then the derived ITA is analyzable with existing techniques and tools.

4.1 Deriving the Pure Markovian Process

Given an IGSMP term $P \in IGSMP_g$ (see [9] or [5] Chapter 7), we derive an IWMC term $Q \in IWMCg$ by approximating general distribution with phase-type distributions.

Since phase-type distributions can be seen as the time to absorption of a continuous time Markov chain, any phase-type distribution *pht* can be represented by some term P_{pht} of IWMC, made up of only weighted prefixes "$w.\_$", exponentially timed prefixes "$\lambda.\_$", choice operators "$\_ + \_$" and occurrences of a variable X representing absorbing states.

Given a function $approx : PDF^+ \multimap PhT$, which associates with each general distribution f occurring in an IGSMP specification P its approximating phase-type distribution *pht*, term $Q \in IWMCg$ is obtained as follows. Denoted with $R[R'/X]$ the term obtained from a term R by replacing R' for X inside R, we just replace each occurrence of a subterm $<f, w>.P'$ in P with $w.(P_{approx(f)}[P'/X])$.

Definition 4.1 Given $P \in IGSMP_g$ and a function $approx : PDF^+ \multimap PhT$, which associates with each general distribution occurring in P an approximating phase-type distribution, we define $\mathcal{M}[\![P, approx]\!] \in IWMCg$ to be the term obtained by replacing each occurrence of a subterm $<f, w>.P'$ in P with $w.(P_{approx(f)}[P'/X])$. ■

The following theorem, where we denote by $approx(P)$ the term of $IGSMP_g$ obtained from $P \in IGSMP_g$ by replacing distributions f in prefixes $<f, w>$

according to *approx*, shows the correctness of the mapping from IGSMP to IWMC terms (performance measures are preserved).

Theorem 4.2 *Given $P \in IGSMP_g$ and $approx : PDF^+ \twoheadrightarrow PhT$, we have that, for every fixed adversary resolving non-deterministic choices, the stochastic process underlying $approx(P)$ is the same as that underlying $\mathcal{M}[\![P, approx]\!]$ (provided that in $\mathcal{M}[\![P, approx]\!]$ we only consider states which do not enable derivatives of terms $P_{approx(f)}$, for any f, as states of the underlying stochastic process).* ■

The following theorem shows that, thanks to the fact that the semantics of IGSMP delays are defined by means of an ST semantics, observational equivalence is preserved when delays are refined by means of phase-type distributions. We denote with $\simeq_{IGSMP}$ observational equivalence over IGSMP terms (defined in [9] or [5] Chapter 7).

Theorem 4.3 *Given $P, Q \in IGSMP_g$ and a function $approx : PDF^+ \twoheadrightarrow PhT$, we have that $P \simeq_{IGSMP} Q$ implies $\mathcal{M}[\![P, approx]\!] \simeq_{IWMC} \mathcal{M}[\![Q, approx]\!]$.*

The simple mapping above from IGSMP terms into IWMC terms is significant from a pure performance viewpoint in that it shows process algebra to provide exactly the machinery necessary for approximating GSMPs with CTMCs through phase-type distributions. This because, while directly transforming at the model level a GSMP into a CTMC via phase-type approximation is really cumbersome due to the interleaving of the exponential phases, when using process algebra we just have to approximate general distributions at the term level and then the parallel operator automatically computes the interleaving of exponential phases for us. Finally, such a mapping confirms ST semantics to be the adequate semantics for generally distributed time in that approximation of activity durations with phase-type distributions is a form of action refinement [1,8].

4.2 Deriving the Pure Real-Time Process

Given an IGSMP $\mathcal{G} = (\Sigma, \mathcal{C}, D, Act, T_+, T_-, T_a, s_0)$ (see [7] or [5] Chapter 6), we derive an ITA $\mathcal{T} = (\Sigma, \mathcal{C}, Act, T_r, T_b, T_a, s_0)$, by turning probabilistic choices into non-deterministic choices and by considering the support of the distribution of a clock as the set of possible values for its duration. In particular, clock start transitions C_i^+ are turned into reset transitions C_i, while clock termination transitions C_i^- are turned into clock bound transitions $C_i \in T$, where T is the support of the distribution $D(C_i)$. Note that a technique like this, which is based on the idea that we introduced in [6] of considering support of distributions as constraints over clocks, was also used in [10] for deriving timed automata from the stochastic automata model of [11]. Subsequently, in [12] it was shown that a more complex technique, which generates new states for each interval composing the domain of the support of the probability distribution of clocks, is actually needed for correctly deriving timed automata from the model of [11]. This because it can be seen that in such a model the direct transformation of clock termination transitions into transitions requiring clocks to assume values in the support of their distributions causes timed automata which behave differently from the original system to be derived. This is due to the fact that in the model of [11] it may happen that a clock termination transition is executed some time after the clock the transition refers to actually terminates. Since such a phenomenon cannot happen in IGSMPs, our simple technique which does not increase the system state space, can be correctly applied.

Now we present the precise definition of support of a probability distribution that we need for the translation. We follow the idea of [12] of defining the support (therein called "useful domain") in such a way that, if a time value is in the support set, then either it has non-zero measure, or it is internal, i.e. it belongs to an open interval which is all included in the support set (and which must have non-zero measure). This avoids considering traces containing action orderings which in the original IGSMP occur with zero probability (see [12]).

Definition 4.4 Given a probability distribution f over $\mathbb{R}$, the support of f, denoted by $supp(f)$, is the set obtained from the least closed subset of $\mathbb{R}$ with measure 1 by eliminating non-internal values with measure 0. ■

It is trivial to verify that for each probability distribution f, $supp(f)$ has measure 1 (hence that the definition is correct).

Definition 4.5 Given an IGSMP $\mathcal{G} = (\Sigma, \mathcal{C}, D, Act, T_+, T_-, T_a, s_0)$, we define $\mathcal{T}[\![\mathcal{G}]\!]$ to be the ITA $(\Sigma, \mathcal{C}, Act, T_r, T_b, T_a, s_0)$, where T_r and T_b are given by

- $T_r = \{(s, C_i, s') \mid (s, C_i^+, s') \in T_+\}$
- $T_b = \{(s, C_i \in T, s') \mid (s, C_i^-, s') \in T_- \wedge T = supp(D(C_i))\}$ ■

In order to show the correctness of the mapping from IGSMP to ITA, we assume the following. Given a state s of an IGSMP and a valuation function v assigning a time value to each of its clocks, we call a "supported execution of an IGSMP starting in (s, v)" a finite sequence of timed transitions $t \in \mathbb{R}^+ \cup 0$ and actions transitions $\alpha \in Act$ executable by the IGSMP according to its semantics (see [7,5]) when it starts in the state s with initial valuation v and when we consider as possible values sampled for a clock with distribution f the time values in $supp(f)$ only. Similarly a "possible execution of an ITA starting in (s, v)" is a finite sequence of timed transitions $t \in \mathbb{R}^+ \cup 0$ and actions transitions $\alpha \in Act$ executable by the ITA according to its semantics (see Appendix B) when it starts in the state s with initial valuation v.

Theorem 4.6 *Given an IGSMP $\mathcal{G} = (\Sigma, \mathcal{C}, D, Act, T_+, T_-, T_a, s_0)$, we have that for each state s and valuation function v associating a time value to the clocks of $\mathcal{G}$ (belonging to $\mathcal{C}$) the set of all supported executions of $\mathcal{G}$ starting in (s, v) is equal to the set of all possible executions of the $\mathcal{T}[\![\mathcal{G}]\!]$ starting in*

(s, v). ■

The following theorem shows that weak bisimulation equivalence is preserved when well-named IGSMPs are mapped into ITA. We denote with $\approx_{IGSMP}$ weak bisimulation over well-named IGSMPs (defined in [9,7] or [5] Chapter 6).

Theorem 4.7 *Given two well-named IGSMPs $\mathcal{G}'$ and $\mathcal{G}''$, we have that $\mathcal{G}' \approx_{IGSMP} \mathcal{G}''$ implies $\mathcal{T}[\![\mathcal{G}']\!] \approx_{ITA} \mathcal{T}[\![\mathcal{G}'']\!]$. Moreover, for each $S, L \subseteq Act - \{\tau\}$, we have $\mathcal{T}[\![\mathcal{G}']\!] \parallel_S \mathcal{T}[\![\mathcal{G}'']\!] \approx_{ITA} \mathcal{T}[\![\mathcal{G}' \parallel_S \mathcal{G}'']\!]$ and $\mathcal{T}[\![\mathcal{G}]\!]/L \approx_{ITA} \mathcal{T}[\![\mathcal{G}/L]\!]$.*

5 Conclusion

In this paper we have presented an idea for an integrated approach for the specification and analysis of stochastic real-time systems based on the usage of probabilistic generally distributed time. Moreover, we have implemented such an approach in the case of interactive systems where specifications are made with the calculus of Interactive Generalized Semi-Markov Processes (IGSMPs) introduced in [9,7,5]. In order to do this we have introduced: the calculus of Interactive Weighted Markov Chains (IWMCs), a pure stochastic-time process algebra, and Interactive Timed Automata, a pure real-time compositional specification paradigm.

As far as future work is concerned, the main goal is to extend the expressiveness of the specification language and to improve usability.

The expressive power of the calculus of IGSMPs, though significant in that it allows internal probabilistic choices, non-determinism and generally distributed probabilistic time to be expressed, should be enhanced in order to increase the capability of modeling real systems with mechanisms like multi-level priorities and interruption of probabilistic time delays. The expressiveness of IWMCs and ITA should then be extended accordingly.

Moreover, once gained an adequate expressive power, the development of a software tool, implementing the transformation techniques introduced and which is interfaced with other standard tools for the analysis of Markov Chains and Timed Automata, would be essential for the actual usability of the integrated approach. Such a tool should also have a friendly graphical user interface to make the development of specifications easy for non experts in process algebra.

References

[1] L. Aceto, M.C.B. Hennessy, *"Adding Action Refinement to a Finite Process Algebra"*, in Information and Computation 115:179-247, 1994

[2] R. Alur, C. Courcoubetis, D. Dill *"Model-Checking in Dense Real-Time"*, in Information and Computation 104:2-34, 1993

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

[4] A. Bobbio, A. Horváth, M. Telek *"The Scale Factor: A New Degree of Freedom in Phase Type Approximation"*, in Proc. of the *Int. Performance & Dependability Symposium (IPDS '02)*, pp. 627-636, Washington (DC), 2002

[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, 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

[8] 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

[9] M. Bravetti, R. Gorrieri, *"The Theory of Interactive Generalized Semi-Markov Processes"*, in Theoretical Computer Science 282: 5-32, 2002

[10] J. Bryans, J. Derrick, *"Stochastic Specification and Verification"*, In Proc. of the *3rd Irish Workshop in Formal Methods*, Electronic Workshops in Computing, July 1999

[11] P.R. D'Argenio, *"Algebras and Automata for Timed and Stochastic Systems"*, Ph.D. Thesis, Univ. Twente, 1997

[12] P.R. D'Argenio, *"A Compositional Translation of Stochastic Automata into Timed Automata"*, Technical Report CTIT 00-08, Univ. Twente, May 2000

[13] P.W. Glynn, *"A GSMP formalism for discrete event simulation"*, in Proc. of the IEEE, 77(1): 14-23, 1989

[14] H. Hansson, B. Jonsson, *"A Framework for Reasoning about Time and Reliability"*, in Proc. of the *10th IEEE Symp. on Real-time Systems*, Santa Monica (CA), 1989

[15] H. Hermanns, *"Interactive Markov Chains"*, Ph.D. Thesis, Universität Erlangen-Nürnberg (Germany), 1998

[16] H. Hermanns, *"An Operator for Symmetry Representation and Exploitation in Stochastic Process Algebras"*, in Proc. of the *5th Workshop on Process Algebras and Performance Modeling*, pp. 55-70, Twente (The Netherlands), 1997

[17] J. Hillston, *"A Compositional Approach to Performance Modelling"*, Cambridge University Press, 1996

[18] K.G. Larsen, A. Skou, *"Bisimulation through Probabilistic Testing"*, in Information and Computation 94:1-28, 1991

[19] R. Milner, *"Communication and Concurrency"*, Prentice Hall, 1989

[20] X. Nicollin, J. Sifakis, *"An Overview and Synthesis on Timed Process Algebras"*, in Real-Time: Theory in Practice, LNCS 600, 1991

[21] X. Nicollin, J. Sifakis, S. Yovine, *"Compiling Real-Time Specifications into Extended Automata"*, in IEEE Trans. on Software Engineering, 18(9):794-804, 1992

[22] M. Scarpa, A. Bobbio, *"Kronecker representation of Stochastic Petri nets with discrete PH distributions"*, in Proc. of the *3rd Int. Performance & Dependability Symposium (IPDS '98)*, pp. 52-61, Durham (NC), 1998

[23] C.M.N. Tofts, *"Processes with Probabilities, Priority and Time"*, in Formal Aspects of Computing 6:536-564, 1994

[24] M.Y. Vardi, *"Automatic Verification of Probabilistic Concurrent Finite-state Programs"*, in Proc. of the *26th IEEE Symp. on Foundations of Computer Science*, pp. 327-338, Portland (OR), 1985

A A Complete Axiomatization for finite state IWMC terms

In this section we present an axiom system which is complete for $\simeq_{IWMC}$ on finite state $IWMC_g$ terms.

The axiom system $\mathcal{A}_{IWMC}$ for $\simeq_{IWMC}$ on $IWMC_g$ terms is formed by the axioms presented in Fig. A.1. In this figure "$\lfloor\!\lfloor$" and "$|$" denote, respectively, the left merge and synchronization merge operators. We recall from Sect. 2 that θ ranges over weights in $\mathbb{R}^+$ and rates in $\mathbb{R}^+$, while $\gamma, \gamma', \ldots$ range over actions in Act, weights and rates.

The axioms $(Pri1)$ and $(Pri2)$ express the two kinds of priorities of $IWMC$, respectively, priority of τ actions over weights and rates and priority of weights over rates. The axiom (Par) is the standard one which expresses parallel composition in terms of left and synchronization merge. The axioms $(Rec1-3)$ handle strongly guarded recursion in the standard way.

If we consider the obvious operational rules for "$\lfloor\!\lfloor_S$" and "$|_S$" that derive from those we presented for the parallel operator [6] then the axioms of $\mathcal{A}_{IWMC}$ are sound.

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

[6] The definition of the operational rule for "$|_S$" must allow for actions "τ" to be skipped, as reflected by axiom $(SM4)$.

$(A1)$	$P + Q = Q + P$	$(A2)$	$(P + Q) + R = P + (Q + R)$
$(A3)$	$\alpha.P + \alpha.P = \alpha.P$	$(A4)$	$P + \underline{0} = P$
$(Tau1)$	$\gamma.\tau.P = \gamma.P$	$(Tau2)$	$P + \tau.P = \tau.P$
$(Tau3)$	$\alpha.(P + \tau.Q) + \alpha.Q = \alpha.(P + \tau.Q)$		
$(Prob)$	$w.P + w'.P = (w + w').P$		
$(ExpT)$	$\lambda.P + \lambda'.P = (\lambda + \lambda').P$		
$(Pri1)$	$\tau.P + \theta.Q = \tau.P$	$(Pri2)$	$w.P + \lambda.Q = w.P$
$(Hi1)$	$\underline{0}/L = \underline{0}$	$(Hi2)$	$(\gamma.P)/L = \gamma.(P/L) \quad \gamma \notin L$
$(Hi3)$	$(a.P)/L = \tau.(P/L) \quad a \in L$	$(Hi4)$	$(P + Q)/L = P/L + Q/L$
$(Rel1)$	$\underline{0}[\varphi] = \underline{0}$	$(Rel2)$	$(\alpha.P)[\varphi] = \varphi(\alpha).(P[\varphi])$
$(Rel3)$	$(\theta.P)[\varphi] = \theta.(P[\varphi])$	$(Rel4)$	$(P + Q)[\varphi] = P[\varphi] + Q[\varphi]$
(Par)	$P \parallel_S Q = P \mathbin{⌊\!⌊}_S Q + Q \mathbin{⌊\!⌊}_S P + P \mid_S Q$		
$(LM1)$	$\underline{0} \mathbin{⌊\!⌊}_S P = \underline{0}$		
$(LM2)$	$(a.P) \mathbin{⌊\!⌊}_S Q = \underline{0} \quad a \in S$		
$(LM3)$	$(\gamma.P) \mathbin{⌊\!⌊}_S Q = \gamma.(P \parallel_S Q) \quad \gamma \notin S$		
$(LM4)$	$(P + Q) \mathbin{⌊\!⌊}_S R = P \mathbin{⌊\!⌊}_S R + Q \mathbin{⌊\!⌊}_S R$		
$(SM1)$	$P \mid_S Q = Q \mid_S P$		
$(SM2)$	$\underline{0} \mid_S P = \underline{0}$		
$(SM3)$	$(\gamma.P) \mid_S (\gamma'.Q) = \underline{0} \quad (\gamma \notin S \ \vee \ \gamma \neq \gamma') \wedge \tau \notin \{\gamma, \gamma'\}$		
$(SM4)$	$(\tau.P) \mid_S Q = P \mid_S Q$		
$(SM5)$	$(a.P) \mid_S (a.Q) = a.(P \parallel_S Q) \quad a \in S$		
$(SM6)$	$(P + Q) \mid_S R = P \mid_S R + Q \mid_S R$		
$(Rec1)$	$recX.P = recY.(P\{Y/X\})$ 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. A.1. Axiomatization for IWMC

Theorem A.1 *If an $IWMC_g$ process P is finite state, then $\exists P' : \mathcal{A}_{IWMC} \vdash P = P'$ with P' sequential state.*

For sequential states the axioms of $\mathcal{A}_{IWMC}$ involved are just standard axioms plus the axioms for priority and probabilistic and exponentially timed choice, hence we have the following.

Theorem A.2 *$\mathcal{A}_{IWMC}$ is complete for $\simeq_{IWMC}$ over finite state $IWMC_g$ processes.*

B A Semantics for Interactive Timed Automata

In Sect. B.1 we introduce Interactive Prioritized Timed Transition Systems (IPTTSes) that will be used in Sect. B.2 to define a semantics for ITA.

B.1 Interactive Prioritized Timed Transition Systems

In this section we formally introduce Interactive Prioritized Timed Transition Systems (IPTTS) which essentially include three type of transitions: *standard action transitions*, representing the interactive behavior of a system component, *prioritized transitions*, representing behaviors of the system component executed according to a certain priority level, and *numeric time transitions* representing a fixed temporal delay.

As far as standard action transitions are concerned they have exactly the same behavior as in ITA. Prioritized transitions are labeled with a certain priority level $p \in \mathbb{N}^+$ and, where transitions with a higher priority level take priority (e.g. when composing two IPTTSes in parallel) over prioritized transitions with a lower priority level. Moreover, we assume standard τ transitions to take priority over prioritized transitions, no matter which is the priority level of such transitions (due to the maximal progress assumption). Given a time domain $TD \subseteq \mathbb{R}^+$, numeric time transitions are labeled with a certain delay $t \in TD$ representing the passage of t time units. As usual in the real time literature (see e.g. [21]), 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 IPTTS we have two different kinds of state:

- *silent states* which are exactly like in ITA.
- *non-silent states* enabling numeric timed transitions and/or prioritized transitions all with the same priority level and (possibly) visible action transitions a, only. In such states numeric timed transitions (which cause the amount of time labeling the transition to pass) and prioritized transitions are chosen by means of a non-deterministic choice. Moreover the IPTTS may potentially interact with the environment through one of its visible actions.

In the following we present the formal definition of Interactive Prioritized Timed Transition System (IPTTS), then we will define Rooted Interactive Prioritized Timed Transition Systems as IPTTSes possessing an initial state. Formally, given a time domain $TD \subseteq \mathbb{R}^+$, we use $t, t', \ldots$, representing time values, to range over TD. Moreover we use $p, p', \ldots$, representing priority levels, to range over $\mathbb{N}^+$. Finally we use θ to range over time values in TD and priorities in $\mathbb{N}^+$.

Definition B.1 An Interactive Prioritized Timed Transition System (IPTTS) is a tuple $\mathcal{D} = (\Sigma, TD, Act, T_p, T_t, T_a)$ with

- Σ a set of possibly infinite 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,
- $T_p \subseteq (\Sigma \times \mathbb{N}^+ \times \Sigma)$ and $T_t \subseteq (\Sigma \times \mathbb{R}^+ \times \Sigma)$ and $T_a \subseteq (\Sigma \times Act \times \Sigma)$ three transition relations representing prioritized behaviors, time passage and action execution, respectively. T_p, T_t and T_a must be such that $\forall s \in \Sigma$.
  - $s \xrightarrow{\ \tau\ } \implies \nexists t. s \xrightarrow{\ t\ } \wedge \ \nexists p. s \xrightarrow{\ p\ }$
  - $s \xrightarrow{\ p\ } \implies \nexists p' < p. s \xrightarrow{\ p'\ }$
  - $s \xrightarrow{\ \tau\ } \vee \exists t. s \xrightarrow{\ t\ } \vee \exists p. s \xrightarrow{\ p\ }$ ■

Definition B.2 A Rooted Interactive Prioritized Timed Transition System (RIPTTS) is a tuple $\mathcal{D} = (\Sigma, TD, Act, T_p, T_t, T_a, s_0)$, where $s_0 \in \Sigma$ is the initial state and $(\Sigma, TD, Act, T_p, T_t, T_a)$ is an IPTTS. ■

The meaning of the constraints over transition relations is the following. The first requirement says that (similarly as in ITA) if a state that can perform internal τ actions then it cannot perform time-related transitions (*maximal progress* assumption). The second requirement says that if a state can perform prioritized transitions with a certain priority level then it cannot perform prioritized transitions with a lower priority level. The third requirement says that (similarly as in ITA) we cannot have states where time is not allowed to pass (time deadlocks).

B.1.1 Parallel of Rooted IPTTSes

Now we define, similarly as for ITA, the parallel composition à la CSP of RIPTTSes.

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

Definition B.3 The parallel composition $\mathcal{D}_1 \,\|_S\, \mathcal{D}_2$ of two RIPTTSes $\mathcal{D}_1 = (\Sigma_1, TD, Act, T_{p,1}, T_{t,1}, T_{a,1}, s_{0,1})$ and $\mathcal{D}_2 = (\Sigma_2, TD, Act, T_{p,2}, T_{t,2}, T_{a,2}, s_{0,2})$, with $S \subset Act - \{\tau\}$ being the synchronization set, is the tuple $(\Sigma, TD, Act, T_p, T_t, T_a, (s_{0,1}, s_{0,2}))$ with:

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

- $T_p \subseteq (\Sigma \times \mathbb{N}^+ \times \Sigma)$, $T_t \subseteq (\Sigma \times TD \times \Sigma)$ and $T_a \subseteq (\Sigma \times Act \times \Sigma)$ the least transition relations, such that

$\mathbf{1_l}$ $s_1 \xrightarrow{\alpha} s'_1,\ \alpha \notin S \implies (s_1, s_2) \xrightarrow{\alpha} (s'_1, s_2)$

$\mathbf{1_r}$ $s_2 \xrightarrow{\alpha} s'_2,\ \alpha \notin S \implies (s_1, s_2) \xrightarrow{\alpha} (s_1, s'_2)$

$\mathbf{2}$ $s_1 \xrightarrow{a} s'_1 \ \wedge\ s_2 \xrightarrow{a} s'_2,\ a \in S \implies (s_1, s_2) \xrightarrow{a} (s'_1, s'_2)$

$\mathbf{3_l}$ $s_1 \xrightarrow{p} s'_1 \ \wedge\ s_2 \overset{\tau}{\not\longrightarrow} \ \wedge\ \not\exists p' > p.s_2 \xrightarrow{p'} \implies (s_1, s_2) \xrightarrow{p} (s'_1, s_2)$

$\mathbf{3_r}$ $s_2 \xrightarrow{p} s'_2 \ \wedge\ s_1 \overset{\tau}{\not\longrightarrow} \ \wedge\ \not\exists p' > p.s_1 \xrightarrow{p'} \implies (s_1, s_2) \xrightarrow{p} (s_1, s'_2)$

$\mathbf{4}$ $s_1 \xrightarrow{t} s'_1 \ \wedge\ s_2 \xrightarrow{t} s'_2 \implies (s_1, s_2) \xrightarrow{t} (s'_1, s'_2)$

- $(s_{0,1}, s_{0,2}) \in \Sigma$ the initial state. ■

When evaluating action transitions we just make use of standard rules. Prioritized transitions are determined by taking into account priorities according to a "global" notion of priority where priorities are applied across the parallel operator. Finally timed transitions are evaluated by just requiring them to synchronize.

Theorem B.4 *Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be two RIPTTSes. Then for each $S \subseteq Act - \{\tau\}$, $\mathcal{D}_1 \|_S \mathcal{D}_2$ is a RIPTTS.* ■

B.1.2 Hiding of Rooted IPTTSes

Now we define, similarly as for ITA, the hiding of RIPTTSes.

Definition B.5 The hiding $\mathcal{D}/L$ of a RIPTTS $\mathcal{D}_1 = (\Sigma, TD, Act, P_1, T_{p,1}, T_{t,1}, T_{a,1}, s_0)$, with $L \subset Act - \{\tau\}$ being the set of visible actions to be hidden, is the tuple $(\Sigma, TD, Act, P, T_p, T_t, T_a, s_0)$, 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
- $T_p \subseteq (\Sigma \times \mathbb{N}^+ \times \Sigma)$, $T_t \subseteq (\Sigma \times TD \times \Sigma)$ and $T_a \subseteq (\Sigma \times Act \times \Sigma)$ the least transition relations, such that $\forall s \in \Sigma$. [7]

$\mathbf{1}$ $s \xrightarrow{\alpha}_1 s',\ \alpha \notin L \implies s \xrightarrow{\alpha} s'$

$\mathbf{2}$ $s \xrightarrow{a}_1 s',\ a \in L \implies s \xrightarrow{\tau} s'$

$\mathbf{3}$ $s \xrightarrow{\theta}_1 \wedge\ \not\exists a \in L.\, s \xrightarrow{a}_1 \implies s \xrightarrow{\theta}$

■

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

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

Theorem B.6 *Let $\mathcal{D}$ be a RIPTTS. Then for each $L \subseteq Act - \{\tau\}$, $\mathcal{D}/L$ is a RIPTTS.* ■

B.1.3 Equivalence of Rooted IPTTSes

Now we introduce a notion of weak bisimulation for RIPTTSes which matches prioritized and timed transitions according to strong bisimulation and abstracts from standard τ actions similarly as in [19].

Definition B.7 Let $\mathcal{D} = (\Sigma, TD, Act, T_p, T_t, T_a)$ be an IPTTS. 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 $\theta \in \mathbb{N}^+ \cup TD$,
$$s_1 \xrightarrow{\ \ \theta\ \ } s_1' \text{ implies } s_2 \stackrel{\theta}{\Longrightarrow} s_2' \text{ for some } s_2' \text{ with } s_1' \ \beta \ s_2',$$

Two states s_1 and s_2 are weakly bisimilar, denoted by $s_1 \approx_{RIPTTS} s_2$, iff (s_1, s_2) is included in some weak bisimulation. Two RIPTTSes $(\mathcal{D}_1, s_{0,1})$ 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 IPTTS obtained with the disjoint union of $\mathcal{D}_1$ and $\mathcal{D}_2$. ■

Theorem B.8 $\approx_{RIPTTS}$ *is a congruence over RIPTTSes whose states may reach via (zero or more) "τ" transitions a state which cannot perform "τ" transitions w.r.t. both parallel and hiding.* ■

B.2 Definition of the Semantics for ITA

In this section we present a semantics for interactive timed automata which maps them onto interactive prioritized timed transition systems. Such a semantics explicitely represents the passage of time by means of transitions labeled with numeric time delays and turns clock reset transitions into prioritized transitions with priority level 2 and clock bound transitions into prioritized transitions with priority level 1.

We now formally define the semantics of an ITA.

Definition B.9 The semantics of an ITA $\mathcal{T} = (\Sigma, \mathcal{C}, Act, T_r, T_b, T_a, s_0)$ is the RIPTTS $[\![\mathcal{T}]\!] = (\Sigma', \mathbb{R}^+ \cup \{0\}, Act, T_p, T_t, T_a', s_0')$ where:

- $\Sigma' = (\Sigma \times Spent)$ is the set of states of the RIPTTS, where *Spent*, ranged over by v, is the set of functions from $\mathcal{C}$ to $\mathbb{R}^+ \cup \{0\}$, expressing the time already spent in execution by the clocks of the ITA from the last reset event
- $\mathbb{R}^+ \cup \{0\}$ is the time domain: we consider continuous time.
- *Act* is the set of standard actions considered in the ITA.
- T_p is the set of prioritized transitions which are defined as the least relation over $\Sigma' \times \mathbb{N}^+ \times \Sigma'$ satisfying the operational rules in the first part of Table B.1.

$(P1)\ \dfrac{s \xrightarrow{\ \phi\ } s' \ \wedge\ v \vdash \phi}{\langle s, v\rangle \xrightarrow{\ 1\ } \langle s', v\rangle}$	$(P2)\ \dfrac{s \xrightarrow{\ C_n\ } s'}{\langle s, v\rangle \xrightarrow{\ 2\ } \langle s', v \leftarrow (C_n, 0)\,\rangle}$
$(T)\ \dfrac{\exists t' \geq t : v + t' \vdash \bigwedge\{\phi \mid s \xrightarrow{\ \phi\ }\}}{\langle s, v\rangle \xrightarrow{\ t\ } \langle s, v + t\rangle}$	
$(A)\ \dfrac{s \xrightarrow{\ \alpha\ } s'}{\langle s, v\rangle \xrightarrow{\ \alpha\ } \langle s', v\rangle}$	

Table B.1
Semantic rules for ITA

- 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 B.1.
- T'_a is the set of action transitions which are defined as the least relation over $\Sigma' \times \mathit{Act} \times \Sigma'$ satisfying the operational rules in the third part of Table B.1.
- $s'_0 = \langle s_0, \mathbf{0}\rangle$, with $\mathbf{0} = \{(C_n, 0) \mid C_n \in \mathcal{C}\}$ is the initial state of the RIPTTS, where the ITA is in the initial state and all clocks start from zero. ∎

In Table B.1 we make use of the following notation. $v \vdash \phi$ holds true if and only if the formula obtained from ϕ by replacing clocks with time values according to v is true. Moreover we define $v \leftarrow (C_n, t)$ to be the function obtained from v by replacing the pair (C_n, t') already contained in v with the new pair (C_n, t). Finally, we define $v + t$, with $t \in \mathbb{R}^+ \cup 0$, to be the function obtained from v by adding t to the time value associated with each clock in v.

Theorem B.10 *Let $\mathcal{T}'$, $\mathcal{T}''$ be two ITA. If $\mathcal{T}' \approx_{ITA} \mathcal{T}''$ then $[\![\mathcal{T}']\!] \approx_{RIPTTS} [\![\mathcal{T}'']\!]$.* ∎

The following theorems show that the semantics of ITA is indeed compositional.

Theorem B.11 *Let $\mathcal{T}'$, $\mathcal{T}''$ be two ITA. For each $S \subseteq \mathit{Act} - \{\tau\}$ we have $[\![\mathcal{T}']\!] \,\|_S [\![\mathcal{T}'']\!] \approx_{RIPTTS} [\![\mathcal{T}' \,\|_S\, \mathcal{T}'']\!]$.* ∎

Theorem B.12 *Let $\mathcal{T}$ be an ITA. For each $L \subseteq \mathit{Act} - \{\tau\}$ we have $[\![\mathcal{T}]\!]/L \approx_{RIPTTS} [\![\mathcal{T}/L]\!]$.* ∎