The How and Why of Interactive Markov Chains*

Holger Hermanns[1,2] and Joost-Pieter Katoen[3,4]

[1] Dependable Systems and Software, Universität des Saarlandes, Germany
[2] VASY Team, INRIA Grenoble – Rhône-Alpes, France
[3] MOVES Group, RWTH Aachen University, Germany
[4] FMT Group, University of Twente, The Netherlands

Abstract. This paper reviews the model of interactive Markov chains (IMCs, for short), an extension of labelled transition systems with exponentially delayed transitions. We show that IMCs are closed under parallel composition and hiding, and show how IMCs can be compositionally aggregated prior to analysis by e.g., bisimulation minimisation or aggressive abstraction based on simulation pre-congruences. We survey some recent analysis techniques for IMCs, i.e., explaining how measures such as reachability probabilities can be obtained. Finally, we demonstrate that IMCs are a natural (and simple) semantic model for stochastic process algebras and generalised stochastic Petri nets and can be used for engineering formalisms such as AADL and dynamic fault trees.

1 Introduction

Designing correct and efficient distributed systems is a difficult task. As a challenging case take an offshore wireless sensor network that is designed to identify tsunami situations and relay tsunami warnings [61]. Once fully operational, will this network help to save human life? Can we guarantee its correct functioning, or is there a risk of failure at the very moment when it is seriously needed? To say it with Barendregt, correct systems for information processing are more valuable than gold [4]. In the tsunami context, a correct system is one that guarantees certain time bounds for the tasks it needs to perform, even in the presence of message losses or component failures. Correctness, performance and dependability are intertwined here, and so they are in many other contemporary IT applications. These applications ask for quantitative correctness properties such as: The frequency of system downtime is below one hour per year, and packets arrive timely in at least 99.96% of all cases.

* This research has been funded by NWO under grant 612.000.420 (QUPES) and DFG-NWO grant Dn 63-257 (ROCKS), by the EU under FP7-ICT-2007-1 grant 214755 (Quasimodo), and by the German Research Council (DFG) as part of the Transregional Collaborative Research Center "Automatic Verification and Analysis of Complex Systems" SFB/TR 14 AVACS.

Performance and dependability evaluation is a discipline that aims at analysing these quantitative system aspects. Major strands of performance evaluation approaches are measurement-based and model-based techniques. In *measurement-based evaluation*, experiments are performed on a concrete (often prototypical) realisation of the system, and timing information is gathered, which is then analysed to evaluate measure(s) of interest. These techniques are routinely practiced in the systems engineering world. They provide specific, precise and very concrete insights into the functioning of a real system. The drawback of these approaches is mainly the fact that they are not reproducible, are hard to scale, and difficult to generalise beyond the concrete setup experimented with. In order to increase reproducibility and reduce costs of larger experiments, distributed systems researchers often resort to *emulation* studies, where the real system code is executed on a virtualised hardware, instead of distributing it physically on the target systems. This especially allows for better concurrency control and thus improved reproducibility. However, it remains notoriously unclear to what extent the imposed control mechanisms tamper the validity of the obtained measures.

In *model-based performance evaluation*, a more general, and thus more abstract approach is taken. A model of the system is constructed that is deemed just detailed enough to evaluate the measure(s) of interest with the required accuracy. In this context the modelling process is an additional step that needs to be performed, and this is a non-trivial task. Process calculi [5] provide a formal basis for designing models of complex systems, especially those involving communicating and concurrently executing components. The underlying basis is the model of labelled transition systems, which represent system behaviour as transitions representing discrete system moves from state to state. The consideration of stochastic phenomena has led to a plethora of stochastic process calculi, cf. the survey in [36]. One of their semantical models is the topic of this paper: *interactive Markov chains* (IMCs, for short) [35]. It stands out in the sense that it extends classical labeled transition systems in a simple yet conservative fashion. IMCs arise from classical concurrency models by incorporating a second type of transitions, denoted $s \xrightarrow{\lambda} s'$, that embodies a random delay governed by a negative exponential distribution with parameter $\lambda \in \mathbb{R}_{>0}$. This twists the model to one that is running on a continuous timeline, and where the execution of actions is supposed to take no time —unless they can be blocked by the environment. (This is linked to the notion of maximal progress.) By dropping the new type of transitions, labeled transition systems are regained in their entirety. By instead dropping the old-fashioned action-labeled transitions, one arrives at one of the simplest but also most widespread class of performance and dependability models, *continuous-time Markov chains* (CTMCs). They can be considered as labeled transition systems, where the transition labels —rates of negative exponential distributions— indicate the speed of the system evolving from one state to another. Their benefits for stochastic process calculi is summarised in [16].

While this simple combination of LTS and CTMCs was at first viewed as a rather academic distinction, the last decade has shown and stressed its importance. First and foremost, IMCs have shown their practical relevance in applica-

tions of various domains, ranging from dynamic fault trees [11,10,12], architectural description languages such as AADL (Architectural Analysis and Design Language) [9,15,13,14], generalised stochastic Petri nets [40] and Statemate [8] to GALS (Globally Asynchronous Locally Synchronous) hardware design [22,19,23]. The availability of CTMC-based tool support [31] for IMCs has led to several of these applications. On the other hand, a rich set of algorithmic advances for the analysis and minimisation of IMCs have been recently developed that enable the analysis of large IMCs [49,66]. Whereas so far the analysis trajectory was restricted to CTMC models obtained from IMCs whose weak bisimulation quotient is free of nondeterminism, with the work of [66] this restriction has become obsolete. In addition, recent developments in compositional abstraction techniques for IMCs are promising means to analyse huge, and even infinite IMCs. This paper provides a survey of IMCs, some of their recent applications and algorithmic advancements.

Organization of this paper. Section 2 introduces IMCs, explains their semantics, defines some basic composition operators and considers (bi)simulation. Section 3 focuses on the analysis of measures-of-interest on IMCs, such as reduction to CTMCs and reachability probabilities of various kinds. Section 4 reports on compositional minimisation techniques for IMCs, including recent progress in aggressive abstraction. Section 5 describes the usage of IMCs as semantical backbone for industrially relevent formalisms such as fault trees and AADL, as well as of other modeling formalisms. Finally, section 6 concludes the paper and gives some propects for future research directions.

2 Interactive Markov chains

What are IMCs? IMCs are basically labeled transition systems with a denumerable state space, action-labeled transitions, as well as Markovian transitions that are labeled with rates of exponential distributions. In the remainder of this paper, we assume the existence of a denumerable set of actions, ranged over by α and β, and which includes a distinguished action, denoted τ. Actions τ models internal, i.e., unobservable activity, whereas all other actions model observable activities.

Definition 1 (Interactive Markov chain). *An* interactive Markov chain *is a tuple* $\mathcal{I} = (S, Act, \longrightarrow, \Longrightarrow, s_0)$ *where*

- S *is a nonempty set of states with* initial state $s_0 \in S$.
- Act *is a set of actions,*
- $\longrightarrow \subseteq S \times Act \times S$ *is a set of* interactive *transitions, and*
- $\Longrightarrow \subseteq S \times \mathbb{R}_{>0} \times S$ *is a set of* Markovian *transitions.*

We abbreviate $(s, \alpha, s') \in \longrightarrow$ as $s \xrightarrow{\alpha} s'$ and similarly, $(s, \lambda, s') \in \Longrightarrow$ by $s \overset{\lambda}{\Longrightarrow} s'$. States are by the type of their outgoing transitions. Let:

- $IT(s) = \left\{ s \xrightarrow{\alpha} s' \right\}$ be the set of interactive transitions that leave s, and

– $MT(s) = \{s \xRightarrow{\lambda} s'\}$ be the set of Markovian transitions that leave s.

A state s is *Markovian* iff $MT(s) \neq \emptyset$ and $IT(s) = \emptyset$; it is *interactive* iff $MT(s) = \emptyset$ and $IT(s) \neq \emptyset$. Further, s is a *hybrid state* iff $MT(s) \neq \emptyset$ and $IT(s) \neq \emptyset$; finally, s is a *deadlock state* iff $MT(s) = IT(s) = \emptyset$. Let $MS \subseteq S$ and $IS \subseteq S$ denote the sets of Markovian and interactive states in IMC $\mathcal{I}$.

A labeled transition system (LTS) is an IMC with $MT(s) = \emptyset$ for any state s. A continuous-time Markov chain (CTMC) is an IMC with $IT(s) = \emptyset$ for any state s. (The case in which $MT(s) = \emptyset = IT(s)$ for any s is both an LTS and a CTMC). IMCs are thus natural extensions of labeled transition systems, as well as of continuous-time Markov chains.

The semantics of an IMC. Roughly speaking, the interpretation of Markovian transition $s \xRightarrow{\lambda} s'$ is that the IMC can switch from state s to s' within d time units with probability $1 - e^{-\lambda \cdot d}$. The positive real value λ thus uniquely identifies a negative exponential distribution. For a Markovian state $s \in MS$, let $\mathbf{R}(s, s') = \sum\{\lambda \mid s \xRightarrow{\lambda} s'\}$ be the *rate* to move from state s to state s'. If $\mathbf{R}(s, s') > 0$ for more than one state s', a competition between the transitions of s exists, known as the *race condition*. The probability to move from such state s to a particular state s' within d time units, i.e., the Markovian transition $s \rightarrow s'$ wins the race, is given by:

$$\frac{\mathbf{R}(s, s')}{E(s)} \cdot \left(1 - e^{-E(s) \cdot d}\right),$$

where $E(s) = \sum_{s' \in S} \mathbf{R}(s, s')$ denotes the *exit rate* of state s. Intuitively, it states that after a delay of at most d time units (second term), the IMC moves probabilistically to a direct successor state s' with discrete branching probability $\mathbf{P}(s, s') = \frac{\mathbf{R}(s,s')}{E(s)}$.

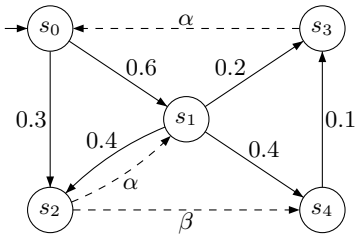


Fig. 1. Example of an IMC with Markovian and interactive states.

Example 1. Consider the IMC $\mathcal{I}$ of Fig. 1 where dotted arrows denote interactive transitions and solid arrows Markovian transitions. We have $MS = \{s_0, s_1, s_4\}$ and $IS = \{s_2, s_3\}$. Markovian states behave like CTMC states, e.g., the transition $s_0 \xRightarrow{0.3} s_2$ expires within $z \in \mathbb{R}_{\geq 0}$ time units with probability $1 - e^{-0.3 \cdot z}$. The

two Markovian transitions of s_0 compete for execution and the transition whose delay expires first is taken. In such a race the sojourn time in s_0 is determined by the first transition that executes. As the minimum of exponential distributions is exponentially distributed with the sum of their rates, the sojourn time of s is determined by its exit rate $E(s)$. In general, the probability to move from a state $s \in MS$ to a successor state $s' \in S$ equals the probability that (one of) the Markovian transitions that lead from s to s' wins the race. Accordingly, $\mathbf{R}(s_0, s_2) = 0.3$, $E(s_0) = 0.3 + 0.6 = 0.9$ and $\mathbf{P}(s_0, s_2) = \frac{1}{3}$. The probability to move from state s_0 to s_2 within 3 time units is $\frac{1}{3} \cdot \left(1 - e^{-2.7}\right)$.

Internal interactive transitions, i.e., τ-labeled interactive transitions, play a special role in IMCs. As they are not subject to any interaction, they cannot be delayed. Thus, internal interactive transitions can be assumed to take place immediately. Now consider a state with both a Markovian transition with rate λ, say, and a τ-transition. Which transition can now occur? As the τ-transition takes no time, it can be taken immediately. The probability that the Markovian transition executes immediately is, however, zero. This justifies that internal interactive transitions take precedence over Markovian transitions. This is called the *maximal progress assumption*.

Definition 2 (Maximal progress). *In any IMC, internal interactive transitions take precedence over Markovian transitions.*

Composition and hiding. The main strength of IMCs is that they are compositional.

Definition 3 (Parallel composition). *Let* $\mathcal{I}_1 = (S_1, Act_1, \longrightarrow_1, \Longrightarrow_1, s_{0,1})$ *and* $\mathcal{I}_2 = (S_2, Act_2, \longrightarrow_2, \Longrightarrow_2, s_{0,2})$ *be IMCs. The* parallel composition *of* $\mathcal{I}_1$ *and* $\mathcal{I}_2$ *wrt. set* A *of actions is defined by:*

$$\mathcal{I}_1 \,||_A\, \mathcal{I}_2 = (S_1 \times S_2, Act_1 \cup Act_2, \longrightarrow, \Longrightarrow, (s_{0,1}, s_{0,2}))$$

where $\longrightarrow$ *and* $\Longrightarrow$ *are defined as the smallest relations satisfying*

1. $s_1 \xrightarrow{\alpha}_1 s_1'$ *and* $s_2 \xrightarrow{\alpha}_2 s_2'$ *and* $\alpha \in A$, $\alpha \neq \tau$ *implies* $(s_1, s_2) \xrightarrow{\alpha} (s_1', s_2')$
2. $s_1 \xrightarrow{\alpha}_1 s_1'$ *and* $\alpha \notin A$ *implies* $(s_1, s_2) \xrightarrow{\alpha} (s_1', s_2)$ *for any* $s_2 \in S_2$
3. $s_2 \xrightarrow{\alpha}_2 s_2'$ *and* $\alpha \notin A$ *implies* $(s_1, s_2) \xrightarrow{\alpha} (s_1, s_2')$ *for any* $s_1 \in S_1$
4. $s_1 \overset{\lambda}{\Longrightarrow}_1 s_1'$ *implies* $(s_1, s_2) \overset{\lambda}{\Longrightarrow} (s_1', s_2)$ *for any* $s_2 \in S_2$
5. $s_2 \overset{\lambda}{\Longrightarrow}_2 s_2'$ *implies* $(s_1, s_2) \overset{\lambda}{\Longrightarrow} (s_1, s_2')$ *for any* $s_1 \in S_1$.

The first three constraints define a TCSP-like parallel composition [45]: actions in A need to be performed by both IMCs simultaneously, except for internal actions (first constraint), whereas actions not in A are performed autonomously (second and third constraint). According to the last two constraints, IMCs can delay independently. This differs from timed models such as timed automata, in which individual processes typically need to synchronise on the advance of time. The memoryless property of exponential distributions justifies independent

delaying: if two Markovian transitions with rates λ and μ, say, are competing to be executed, then the remaining delay of the μ-transition after the λ-transition has been taken, is exponentially distributed with rate μ.

Definition 4 (Hiding). *The* hiding *of IMC $\mathcal{I} = (S, Act, \longrightarrow, \Longrightarrow, s_0)$ wrt. the set A of actions is the IMC $\mathcal{I} \setminus A = (S, Act \setminus A, \longrightarrow', \Longrightarrow, s_0)$ where $\longrightarrow'$ is the smallest relation defined by:*

1. *$s \xrightarrow{\alpha} s'$ and $\alpha \notin A$ implies $s \xrightarrow{\alpha}' s'$, and*
2. *$s \xrightarrow{\alpha} s'$ and $\alpha \in A$ implies $s \xrightarrow{\tau}' s'$.*

Hiding thus transforms α-transitions with $\alpha \in A$ into τ-transitions. All other transition labels remain unaffected. This operation is of importance for the maximal progress assumption of IMCs. Turning an α-transition emanating from state s, say, into a τ-transition may change the semantics of the IMC at hand, as after hiding no Markovian transition will be ever taken in s.

Bisimulation. To compare IMCs, we introduce the notions of strong and weak bisimulation. For set $C \subseteq S$ of states and state s, let $\mathbf{R}(s, C) = \sum_{s' \in C} \mathbf{R}(s, s')$. Intuitively, two states s and t are strongly bisimilar if any interactive transition $s \xrightarrow{\alpha} s'$ can be mimicked by t, i.e., $t \xrightarrow{\alpha} t'$ such that s' and t' are bisimilar. In addition, the cumulative rate of moving from s to some equivalence class C of states, i.e., $\mathbf{R}(s, C)$ equals $\mathbf{R}(t, C)$. Since the probability of a Markovian transition to be executed immediately is zero, whereas internal interactive transitions take always place immediately, there is no need to require equality of cumulative rates if states have outgoing internal transitions. Let $s \xrightarrow{\tau}\!\!\!\!\!/\;\;$ denote a predicate that is true if and only if s has no outgoing τ-transition. For state s, action α and $C \subseteq S$, let $\mathbf{T}(s, \alpha, C) = 1$ if and only if $\{s' \in C \mid s \xrightarrow{\alpha} s'\}$ is non-empty.

Definition 5 (Strong bisimulation). *Let $\mathcal{I} = (S, Act, \longrightarrow, \Longrightarrow, s_0)$ be an IMC. An equivalence relation $R \subseteq S \times S$ is a* strong bisimulation *on $\mathcal{I}$ if for any $(s, t) \in R$ and equivalence class $C \in S/R$ the following holds:*

1. *for any $\alpha \in Act$, $\mathbf{T}(s, \alpha, C) = \mathbf{T}(t, \alpha, C)$, and*
2. *$s \xrightarrow{\tau}\!\!\!\!\!/\;\;$ implies $\mathbf{R}(s, C) = \mathbf{R}(t, C)$.*

States s and s' are strongly bisimilar, denoted $s \sim s'$, if $(s, s') \in R$ for some strong bisimulation R.

The rate equality is adopted from the notion of lumping equivalence [18]. Two IMCs $\mathcal{I}_1$ and $\mathcal{I}_2$ on (disjoint) state spaces S_1 and S_2 respectively are bisimilar, denoted $\mathcal{I}_1 \sim \mathcal{I}_2$, if there exists a strong bisimulation R on $S_1 \cup S_2$ such that $(s_{0,1}, s_{0,2}) \in R$. The next property asserts that $\sim$ is substitutive with respect to parallel composition and hiding, so, e.g., $\mathcal{I} \sim \mathcal{I}'$ implies for any set A that $\mathcal{I} \setminus A \sim \mathcal{I}' \setminus A$.

Theorem 1. *[35] $\sim$ is a congruence wrt. parallel composition and hiding.*

As discussed before, τ-transitions play a special role in IMCs. Whereas strong bisimulation treats all interactive transitions in the same way, regardless whether they are internal (i.e., labelled by τ) or not, weak bisimulation takes an observer's point of view and cannot distinguish between executing several successive τ-transitions or a single one. This allows for collapsing sequences of internal interactive transitions by a single such transition. This acts exactly the same as for labeled transition systems. The treatment of Markovian transitions is a bit more involved, however. First, let us remark that the probability distribution of a sequence of exponential distributions is not an exponential distribution but constitutes a phase-type distribution. Therefore, it is not possible to define a weak version of the transition relation $\Longrightarrow$ as is done for weak bisimulation in labeled transition systems. The solution is to demand that Markovian transitions have to be mimicked in the strong sense, while they can be preceded and/or followed by arbitrary sequences of internal interactive transitions. The treatment of sequences of internal interactive transitions is similar to that of branching bisimulation [62]. As for strong bisimulation, rate equality is only required if a state has no outgoing internal transitions (maximal progress). Let $s \xrightarrow{\tau^*} s'$ denote that s' can be reached from s solely via zero or more τ-transitions; in particular $s \xrightarrow{\tau^*} s$ for any state s. For state s, action α and $C \subseteq S$, let $\mathbf{W}(s, \alpha, C) = 1$ if and only if $\{s' \in C \mid s \xrightarrow{\tau^*} \xrightarrow{\alpha} \xrightarrow{\tau^*} s'\}$ is non-empty.

Definition 6 (Weak bisimulation). *Let $\mathcal{I} = (S, Act, \longrightarrow, \Longrightarrow, s_0)$ be an IMC. An equivalence relation $R \subseteq S \times S$ is a* weak bisimulation *on $\mathcal{I}$ if for any $(s,t) \in R$ and equivalence class $C \in S/R$, the following holds:*

1. *for any $\alpha \in Act$, $\mathbf{W}(s, \alpha, C) = \mathbf{W}(t, \alpha, C)$, and*
2. *$s \xrightarrow{\tau^*} s'$ and $s' \not\xrightarrow{\tau}$ implies $t \xrightarrow{\tau^*} t'$ and $t' \not\xrightarrow{\tau}$ and $\mathbf{R}(s', C) = \mathbf{R}(t', C)$ for some $t' \in S$.*

States s and s' are weakly bisimilar, denoted $s \approx s'$, if $(s, s') \in R$ for some weak bisimulation R.

Theorem 2. *[35] $\approx$ is a congruence wrt. parallel composition and hiding.*

Bisimulation relations are equivalences requiring two bisimilar states to exhibit identical stepwise behaviour. On the contrary, simulation relations [46] are preorders on the state space requiring that whenever $s \preceq s'$ (s' simulates s) state s' can mimic all stepwise behaviour of s; the converse is not guaranteed, so state s' may perform steps that cannot be matched by s.

Definition 7 (Strong simulation). *For IMC $\mathcal{I} = (S, Act, \longrightarrow, \Longrightarrow, s_0)$, $R \subseteq S \times S$ is a* simulation relation*, iff for any $(s,t) \in R$ it holds:*

1. *for any $\alpha \in Act$ and $s' \in S$, $s \xrightarrow{\alpha} s'$ implies $t \xrightarrow{\alpha} t'$ and $(s', t') \in R$ for some $t' \in S$*
2. *$s \not\xrightarrow{\tau}$ implies $E(s) \leq E(t)$*
3. *$s \not\xrightarrow{\tau}$ implies for distributions $\mu = \mathbf{P}(s, \cdot)$ and $\mu' = \mathbf{P}(s', \cdot)$ there exists $\Delta : S \times S \to [0,1]$ such that for all $u, u' \in S$:*

$$\text{(a) } \Delta(u,u') > 0 \implies (u,u') \in R \quad \text{(b) } \Delta(u,S) = \mu(u) \quad \text{(c) } \Delta(S,u') = \mu'(u')$$

We write $s \preceq s'$ if $(s,s') \in R$ for some simulation R and $\mathcal{I} \preceq \mathcal{I}'$ for IMCs $\mathcal{I}$ and $\mathcal{I}'$ with initial states s_0 and s_0', if $s_0 \preceq s_0'$ in the disjoint union of $\mathcal{I}$ and $\mathcal{I}'$. The last constraint requires the existence of a weight function Δ that basically distributes μ of s to μ' of s' such that only related states obtain a positive weight (3(a)), and the total probability mass of u that is assigned by Δ coincides with $\mu(u)$ and symmetrically for u' (cf. 3(b), 3(c)).

Theorem 3. *$\preceq$ is a precongruence wrt. parallel composition and hiding.*

Constraint-oriented specification of performance aspects. Let us conclude this section by describing how IMCs can be used to meet the challenges as put forward in the well-known paradigm of separation of concerns. We do so by showing that IMCs can be naturally used to specify performance aspects in the so-called constraint-oriented specification style [64]. This style is a format par excellence to support the separation of concerns principle when specifying the characteristics of complex distributed systems. It has been originally developed to support the early phases of the design trajectory. Put in a nutshell, constraints are viewed as separate processes. Parallel composition is used to combine these constraints much in the same vein as logical conjunction.

To illustrate how IMCs perfectly match the constraint-oriented specification style consider a given system model P that does not contain random timing constraints yet —i.e., P is a labeled transition system— and let α and β be two successive actions in P. To insert a random delay between these two actions, it now suffices to construct an IMC D_p with an initial state with outgoing transition α and a final state, i.e. a state without outgoing transitions, that can can only be reached by a β-transition. The state reached after performing α and the state from which the β-transition is emanating are connected by a CTMC, i.e., an IMC with only Markovian transitions. This CTMC models the random delay that we want to impose on the delay between α and β. The resulting system is now obtained as $P \,||_{\{\alpha,\beta\}}\, D_p$. The "delay" process D_p is thus imposed as additional constraint to process P. This procedure can now be repeated to impose delays between other actions in P. As CTMCs can approximate general probability distributions arbitarily closely, this is a powerful recipe. This is exemplified in [39] where a complex telephone system specification in LOTOS has been enriched with performance characteristics using a constraint-oriented specification style.

Now assume that we want to impose random delays on some of the observable actions from P and Q. Following the procedure just described, this yields

$$(P \,||_A\, Q) \;||_{A_p \cup A_q}\; (D_p \,||_\emptyset\, D_q)$$

where A_p are the synchronised actions with "delay" process D_p and A_q the ones with D_q. Note that the timing constraints are added "on top" of the entire specification. As it suffices to impose a single delay on each action, the processes D_p and D_q are independent, and thus need not to synchronise. In case D_p

delays some local actions from P, and D_q delays local actions from Q, the above specification can be rewritten into the weak bisimilar specification:

$$\underbrace{\left(P \,||_{A_p}\, D_p\right)}_{\text{local constraints of } P} \quad ||_A \quad \underbrace{\left(Q \,||_{A_q}\, D_q\right)}_{\text{local constraints of } Q}$$

Note that in this system specification, the functional and performance aspects of each individual component are separated, as well as the specifications of the components themselves.

3 IMC analysis

Assume that the IMC under consideration is complete, i.e., it is not subject any further to interaction with other components that are modeled as IMCs. This is important, as this means that actions cannot be further delayed due to a delay which is imposed by the environment. Formally, this means that we can safely hide all actions in the IMC at hand, i.e., we consider $\mathcal{I} \setminus A$ where A contains all actions occuring in $\mathcal{I}$. Accordingly, all actions are labeled by τ. The typical specification that is subject to analysis is thus of the form:

$$(\mathcal{I}_1 \,||_{A_1}\, \mathcal{I}_2 \,||_{A_2}\, \ldots \,||_{A_N}\, \mathcal{I}_N) \setminus A$$

where A is the union of all actions in IMC $\mathcal{I}_i$, i.e., $A = \cup_{i=1}^{N} Act_i$. Due to the maximal progress assumption, the resulting IMC can be simplified: in any state that has a τ-transition, all Markovian transitions can be removed. Subsequently, sequences of τ-transitions can be collapsed by applying weak bisimulation. If nondeterminism is absent in the resulting IMC, in fact a CTMC remains, and all analysis techniques for CTMCs can be employed [34], such as transient or steady-state analysis or CSL model checking [2].

Time-bounded reachability. An alternative analysis technique is to compute time-bounded reachability probabilities. This does not require the IMC to be reducible to a CTMC, and can thus be applied to *any* IMC. Let us explain the kind of measure we are interested in. First, consider infinite paths in an IMC. An infinite path π in an IMC is an infinite sequence of the form

$$\pi \;=\; s_0 \xrightarrow{\sigma_0,t_0} s_1 \xrightarrow{\sigma_1,t_1} s_2 \xrightarrow{\sigma_2,t_2} \ldots$$

with $s_i \in S$, σ_i is either an action in Act or equals $\bot$, and $t_i \in \mathbb{R}_{\geq 0}$. The occurrence of action α after a delay of t time units in state s_i in π is denoted by $s_i \xrightarrow{\alpha,t} s_{i+1}$; in case of a Markovian transition after t time units delay, this is denoted by $s_i \xrightarrow{\bot,t} s_{i+1}$. As internal interactive transitions take place immediately, their occurrence is denoted $s_i \xrightarrow{\tau,0} s_{i+1}$. For time point $t \in \mathbb{R}_{\geq 0}$, let $\pi@t$ denote the sequence of states that π occupies at time t. Note that $\pi@t$ is in general not a single state, but rather a sequence of several states, as an IMC may exhibit

immediate transitions and thus may occupy various states at the same time instant. An example path in the IMC of Fig. 1 is $s_0 \xrightarrow{\bot,3.0} s_1 \xrightarrow{\bot,2.0} s_2 \xrightarrow{\beta,0} s_4 \ldots$ which occupies the states s_2 and s_4 at time instant 5.0. Let $Paths^{\omega}(s)$ denote the set of infinite paths starting in state s. Using a standard cylinder construction, a sigma-algebra can be defined over the set of infinite paths of an IMC, and can be equipped with a probability measure [66], denoted Pr in the sequel.

Now, let $\mathcal{I}$ be an IMC with state space S, initial state s, and let $G \subseteq S$ be a set of goal states and $I \subseteq \mathbb{R}$ a time interval with rational bounds. The time-bounded reachability event $\Diamond^I G$ is defined as:

$$\Diamond^I G \;=\; \{\pi \in Paths^{\omega}(s) \mid \exists t \in I.\, \exists s' \in \pi@t.\, s' \in G\}$$

It thus contains all infinite paths starting in state s that hit a state in G at some time point that lies in the interval I. We are basically interested in the probability of the event $\Diamond^I G$. The problem, however, is that —due to the presence of non-determinism— this is not uniquely defined. To see this, consider the IMC of Fig. 1 with $G = \{s_4\}$. The probability of the event $\Diamond^{[0,2]} G$ for state s_2, for instance, now depends on how the non-deterministic choice between α and β has been resolved in state s_2. If β is chosen the probability equals one; otherwise it depends on the choice in state s_1. We therefore consider the probability of $\Diamond^I G$ relative to a specific resolution of the non-determinism in the IMC. Such resolution is defined by a total-time deterministic positional *policy* D, say. It goes beyond the scope of this paper to fully define this class of policies. For the sake of the remainder of this paper, it suffices to consider D as a function that takes as argument the current state s_i, say, and the total time that has elapsed along the path leading to s_i, including the time already spent in state s_i so far. Based on this information, D will select one of the actions of an outgoing transition of s_i.

Example 2. Consider again the IMC of Fig. 1. Assume the execution of the IMC so far is $s_0 \xrightarrow{\bot,3.0} s_1 \xrightarrow{\bot,2.0} s_2$. A choice between the actions α and β has to be made in s_2. An example policy D is $D(s_2, t) = \alpha$ if $t \leq 10$, and $D(s_2, t) = \beta$ otherwise. Thus, if the residence time in the current state s_2 is d time units, say, then α will be chosen if $d \leq 5$ (as 5 time units have passed until reaching s_2), whereas β will be chosen if $d > 5$.

We can now be more precise about the measure-of-interest: we are interested in maximizing the probability of $\Diamond^I G$ for all possible total-time dependent policies, i.e., we want to determine

$$p^{\max}(s, I) \;=\; \sup_{D} \Pr_{s,D}\left(\Diamond^I G\right) \qquad \text{for timed policy } D.$$

One may wonder whether we should not consider more powerful classes of policies, such as randomised ones, or policies that may base their decision on the entire computation so far, but this does not lead to a larger value for $p^{\max}(s, I)$:

Theorem 4. *[57] Total-time deterministic positional policies are optimal for maximising* $\Pr(\Diamond^I G)$.

Reachability probabilities. Before discussing how to compute $p^{\max}(s, I)$, let us first discuss a simpler variant of the event $\Diamond^I G$. This case occurs if $\sup I = \infty$ and $\inf I = 0$. As the time interval does not impose any timing constraint anymore, this amounts to a simple reachability event:

$$\Diamond G \;=\; \{\pi \in Paths^{\omega}(s) \mid \exists i \in \mathbb{N}.\, \pi[i] \in G\}$$

where $\pi[i]$ denotes the i-th state along π. Thus all paths are considered that hit G at some position, no matter how much time has elapsed upon hitting G. For such (unbounded) reachability events, positional policies suffice, i.e., there is no need anymore to "know" the total time that has elapsed along the computation so far. In fact, $p^{\max}(s, [0, \infty))$ can be determined by considering the discrete-probabilistic process that is embedded in the IMC at hand. The discretised counterpart of an IMC is an interactive probabilistic chain.

Definition 8 (Interactive probabilistic chain [23]). *An* interactive probabilistic chain *(IPC) is a tuple $\mathcal{P} = (S, Act, \longrightarrow, \mathbf{P}, s_0)$, where S, Act, IT and s_0 are as in Def. 1 and $\mathbf{P} : S \times S \to [0, 1]$ is a transition probability function sastifying $\forall s \in S.\, \mathbf{P}(s, S) \in \{0, 1\}$.*

A state s in an IPC $\mathcal{P}$ is *probabilistic* iff $\sum_{s' \in S} \mathbf{P}(s, s') = 1$ and $IT(s) = \emptyset$. As for IMCs, we adopt the *maximal progress assumption*. Hence, interactive internal transitions take precedence over probabilistic transitions and their execution takes zero discrete time steps. The embedded IPC of an IMC is obtained by considering the discrete-probabilistic interpretation of $\Longrightarrow$, i.e., $\mathbf{P}(s, s') = \frac{\mathbf{R}(s,s')}{E(s)}$ if $MT(s) \neq \emptyset$, and 0 otherwise. It then follows:

Theorem 5. *For any IMC $\mathcal{I}$ with embedded IPC $\mathcal{P}$: $p^{\mathcal{I}}(s, [0, \infty)) = p^{\mathcal{P}}(s, [0, \infty))$.*

The values $p^{\mathcal{P}}(s, [0, \infty))$ can be obtained by applying a slight variation of value iteration algorithms for MDPs [7].

Discretisation. The computation of $p^{\max}(s, I)$ with $\inf I \neq \emptyset$ can be done using discretisation, and as we will see, can also be reduced —though in a different way as explained above— to value iteration on MDPs.

Definition 9 (Discretisation [66]). *An IMC $\mathcal{I} = (S, Act, \longrightarrow, \Longrightarrow, s_0)$ and a step duration $\delta \in \mathbb{R}_{>0}$ induce the* discretised IPC *$\mathcal{P}_\delta = (S, Act, \longrightarrow, \mathbf{P}', s_0)$, where*

$$\mathbf{P}'(s, s') = \begin{cases} \left(1 - e^{-E(s)\cdot\delta}\right) \cdot \mathbf{P}(s, s') & \text{if } s \neq s' \\ \left(1 - e^{-E(s)\cdot\delta}\right) \cdot \mathbf{P}(s, s') + e^{-E(s)\cdot\delta} & \text{if } s = s'. \end{cases} \tag{1}$$

Let $p^{\mathcal{P}}_{max}(s, [k_a, k_b])$ for an IPC $\mathcal{P}$ with state s and step-interval $0 \leq k_a \leq k_b$ be the supremum of the probabilities to reach a set of goal states within step interval $[k_a, k_b]$, $k_a, k_b \in \mathbb{N}$. The following result allows to approximate this probability in the underlying IMC by a step-bounded reachability analysis in its discretised IPC. This discretisation is indeed *quantifiably correct*:

Theorem 6 (Approximation theorem [66]). *Let $\mathcal{I} = (S, Act, \longrightarrow, \Longrightarrow, s_0)$ be an IMC, $G \subseteq S$ a set of goal states and $\delta > 0$ a step duration. Further, let I be a time interval with $\inf I = a$ and $\sup I = b$ such that $a < b$ and $a = k_a\delta$ and $b = k_b\delta$ for some $k_a \in \mathbb{N}$ and $k_b \in \mathbb{N}_{>0}$. Then:*

$$p^{\mathcal{P}_\delta}_{max}\big(s, (k_a, k_b]\big) - k_a \cdot \frac{(\lambda\delta)^2}{2} \leq p^{\mathcal{I}}_{max}(s, I) \leq p^{\mathcal{P}_\delta}_{max}\big(s, (k_a, k_b]\big) + k_b \cdot \frac{(\lambda\delta)^2}{2} + \lambda\delta.$$

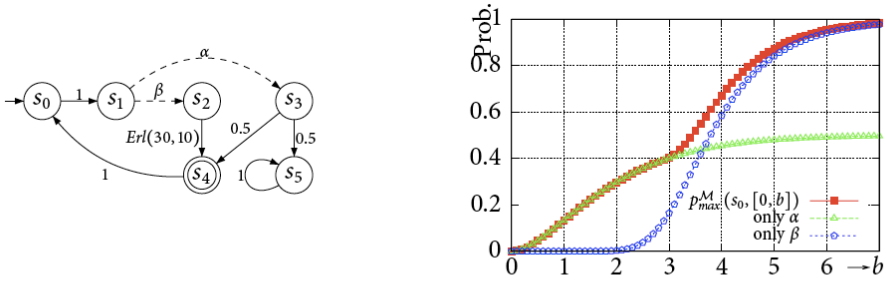
Given an error bound ε, we can choose a sufficiently small step duration $\delta > 0$ such that $\left|p^{\mathcal{P}_\delta}_{max}\big(s, (k_a, k_b]\big) - p^{\mathcal{I}}_{max}(s, I)\right| \leq k_b \cdot \frac{(\lambda\delta)^2}{2} + \lambda\delta < \varepsilon$ holds. Note that this can be done *a priori*. Hence, $p^{\mathcal{P}_\delta}_{max}\big(s, (k_a, k_b]\big)$ approximates the probabilities $p^{\mathcal{I}}_{max}(s, I)$ up to ε. Further, $p^{\mathcal{P}_\delta}_{max}\big(s, (k_a, k_b]\big)$ can easily be computed by slightly adapting the well-known value iteration algorithm for MDPs [7]. For an error-bound $\varepsilon > 0$ and a time-interval I with $\sup I = b$, this approach has a worst case time complexity in $\mathcal{O}\big(n^{2.376} + (m + n^2) \cdot (\lambda b)^2 / \varepsilon\big)$ where λ is the maximal exit rate and m and n are the number of transitions and states of the IMC, respectively.

Example 3. (Adopted from [58].) Consider the IMC depicted in Fig. 2(a). Let $G = \{s_4\}$ as indicated by the double-circled state s_4. The only state which exhibits non-determinism is state s_1 where a choice between α and β has to be made. Selecting α rapidly leads to the goal state as with probability $\frac{1}{2}$, s_4 is reached with an exponential distribution of rate one. Selecting β almost surely leads to the goal state, but, however, is subject to a delay that is governed by an Erlang(30,10)-distribution, i.e., a sequence of 30 exponential distributions of each rate 10. Note that this approximates a deterministic delay of 3 time units. The time-bounded reachability probabilities are plotted in Fig 2(b). This plot clearly shows that it is optimal to select α upto about time 3, and β afterwards. The size of the IMC, its maximal exit rate (λ), accuracy (ϵ), time bound (b) and the computation time are indicated in Fig. 2(c).

Time-bounded reachability-avoid probabilities. To conclude this section, we will explain that determining $p^{\max}(s, I)$ can also be used for more advanced measures-of-interest, such as "reach-avoid" probabilities. Let, as before, s be a state in an IMC, $I = [0, d]$ a time interval with rational d, G be a set of goal states, and A a set of states that need to be avoided before reaching G. The measure-of-interest now is to maximise the probability to reach G at some time point in the interval I while avoiding any state in A prior to reaching G. Formally, the event-of-interest is:

$$\overline{A}\,\mathsf{U}^{[0,d]}\,G \;=\; \{\pi \in Paths^{\omega}(s) \mid \exists t \leq d.\, \exists s' \in \pi@t.\, s' \in G \wedge \forall s'' \in pref(s').\, s'' \notin A\}$$

where $pref(s')$ is the set of states along π that are reached before reaching s' and $\overline{A}$ is the complement of A, i.e., $\overline{A} = S \setminus A$. The maximal probability of this event can be computed in the following way. The IMC is first transformed by making all states in G absorbing, i.e., for any state $s \in G$, the outgoing transitions are removed. This is justified by the fact that it is not of importance what happens once a state in G has been reached (via a $\overline{A}$-path); in addition, if a G-state is



(a) The $Erl(30, 10)$ model $\mathcal{M}$.

(b) Time-bounded reachability in $\mathcal{M}$.

problem	states	ε	λ	b	prob.	time
$Erl(30,10)$	35	10^{-3}	10	4	0.672	50*s*
$Erl(30,10)$	35	10^{-3}	10	7	0.983	70*s*
$Erl(30,10)$	35	10^{-4}	10	4	0.6718	268*s*

(c) Computation times for different parameters.

Fig. 2. Time-bounded reachability probabilities in an example IMC

reached before the deadline d, this does not matter, as it will still be in G at time d since it is made absorbing. In addition, all states in $A \cap \overline{G}$ are made absorbing as the probability of a path that reaches an A-state which is also a $\overline{G}$-state to satisfy the event-of-interest is zero. The resulting structure is thus an IMC in which only the states in $\overline{A} \setminus G$ are unaffected; all other states are made absorbing. It now follows in a similar way as in [2]:

Theorem 7. $\underbrace{\sup_D \Pr_{s,D}\left(\overline{A}\,\mathsf{U}^{[0,d]}\,G\right)}_{\text{in the IMC } \mathcal{I}} = \underbrace{\sup_D \Pr_{s,D}\left(\Diamond^{[0,d]}\,G\right)}_{\text{in the IMC } \mathcal{I}'}.$

Here, IMC $\mathcal{I}'$ is obtained from $\mathcal{I}$ by making all states outside $\overline{A} \setminus G$ absorbing. As a result of the above theorem, computing time-bounded reach-avoid probabilities is reduced to determining time-bounded reachability probabilities, which can be determined in the way described earlier. It goes without saying that a similar strategy can be applied to (unbounded) reach-avoid probabilities.

4 Abstraction

As for any state-based technique, the curse of dimensionality is a major limitation for IMCs. Although its approximate analysis algorithms as described above are polynomial (with relatively low degree) in the state space size, state spaces of realistic systems easily consist of millions or even billions of states. In order to deal with such systems, aggressive abstraction techniques are required. In the following, we consider abstraction techniques that are based on partitioning

the state space into groups of states. A possibility to achieve this, is to apply bisimulation minimisation.

Compositional bisimulation minimisation. An important property that provides the basis for such abstraction is the fact that for bisimilar states time-bounded (as well as unbounded) reachability probabilities are preserved:

Theorem 8. *[56] For any finitely-branching IMC with state space S, states $s, s' \in S$, $G \subseteq S$ and time interval I:*

$$s \sim s' \quad \textit{implies} \quad p^{\max}(s, I) \;=\; p^{\max}(s', I).$$

The above result opens the way to generate —prior to any (time-consuming) analysis— an IMC that is bisimilar to the IMC under consideration, but preferably much smaller. This is called the quotient IMC. For equivalence relation R on state space S and $s \in S$, let $[s]_R$ denote the equivalence class of s under R, and let $S/R = \{[s]_R \mid s \in S\}$ denote the quotient space of S under R.

Definition 10 (Quotient IMC). *Let $\mathcal{I} = (S, Act, \longrightarrow, \Longrightarrow, s_0)$ be an IMC and R a strong bisimulation on S. The* quotient IMC $\mathcal{I}/R = \big(S/R, Act, \longrightarrow', \Longrightarrow', [s_0]_R\big)$ *where $\longrightarrow'$ and $\Longrightarrow'$ are the smallest relations satisfying:*

1. $s \xrightarrow{\alpha} s'$ *implies* $[s]_R \xrightarrow{\alpha}{}' [s']_R$ *, and*
2. $s \overset{\lambda}{\Longrightarrow} s'$ *implies* $[s]_R \overset{\mathbf{R}(s,[s']_R)}{\Longrightarrow}{}' [s']_R$.

It now follows that for any IMC $\mathcal{I}$ and strong bisimulation, it holds $\mathcal{I} \sim \mathcal{I}/R$. (A similar result holds for weak bisimulation, replacing $\sim$ by $\approx$).

The next question is how to obtain the bisimulation quotient of a given IMC, and preferably even the quotient with respect to the coarsest bisimulation, as this yields an IMC of minimal size which is strong bisimilar to the original one. Using a variant of Paige-Tarjan's partition-refinement algorithm for computing strong bisimulation on labeled transition systems we obtain:

Theorem 9. *[35] For any IMC $\mathcal{I}$ with state space S and strong bisimulation R on S, the quotient IMC $\mathcal{I}/R$ can be computed in time complexity $\mathcal{O}(m \log n)$ where m and n are the number of transitions and states of the IMC $\mathcal{I}$.*

The results so far suggest to compute the quotient IMC prior to the analysis of, e.g., time-bounded reachability probabilities. This leads to significant state-space reductions and efficiency gains in computation times, as e.g., is shown in [47] for CTMCs. But, as the bisimulation minimisation is not an on-the-fly algorithm, it requires the entire state space of the original, i.e., non-minimised IMC up front. For realistic systems, this requirement is a significant burden. Fortunately, as IMCs are compositional —they can be put in parallel in a simple manner— and as bisimulation is a congruence wrt. parallel composition, bisimulation minimisation can be applied in a *component-wise manner*. This works as follows. Suppose the system-to-be-analysed is of the form:

$$\mathcal{I} \;=\; \mathcal{I}_1 \,||_{A_1}\, \mathcal{I}_2 \,||_{A_2}\, \ldots \,||_{A_N}\, \mathcal{I}_N,$$

i.e., a parallel composition of N IMCs. For the sake of our argument, let us assume that the size of $\mathcal{I}$ is too large to be handled, and therefore bisimulation minimisation cannot be applied. However, each component is of a moderate size that can be subject to minimisation. Let $\widehat{\mathcal{I}_i}$ be the quotient of IMC $\mathcal{I}_i$, for $0 < i \leq N$. Each such quotient can be obtained by the aforementioned partition-refinement algorithm. Thanks to the property that bisimulation is substitutive wrt. parallel composition, it follows from the fact that $\mathcal{I}_i \sim \widehat{\mathcal{I}_i}$, for $0 < i \leq N$, that:

$$\mathcal{I}_1 \,||_{A_1}\, \mathcal{I}_2 \,||_{A_2}\, \ldots \,||_{A_N}\, \mathcal{I}_N \;\sim\; \widehat{\mathcal{I}_1} \,||_{A_1}\, \widehat{\mathcal{I}_2} \,||_{A_2}\, \ldots \,||_{A_N}\, \widehat{\mathcal{I}_N}.$$

The worst case time complexity to obtain this reduced system is determined by the largest IMC $\mathcal{I}_i$ and equals $\mathcal{O}(\max_i(m_i \log n_i))$ where m_i and n_i are the number of transitions and states in IMC $\mathcal{I}_i$. Similar reasoning applies to weak bisimulation, with the exception that the time complexity for determining the quotient under weak bisimulation requires the computation of a transitive closure which is in $\mathcal{O}(n^{2.376})$. As weak bisimulation also preserves maximal time-bounded reachability probabilities, and is substitutive, an IMC can be minimised compositionally before any analysis:

Theorem 10. *For any finitely-branching IMC with state space S, states $s, s' \in S$, $G \subseteq S$ and time interval I:*

$$s \approx s' \quad \textit{implies} \quad p^{\max}(s, I) \;=\; p^{\max}(s', I).$$

Finally, for simulation preorders we obtain a slightly other preservation result. Intuitively speaking, whenever $\mathcal{I} \preceq \mathcal{I}'$, then $\mathcal{I}'$ can mimic all behaviours of $\mathcal{I}$, but perhaps can do more (and faster). This yields:

Theorem 11. *For any finitely-branching IMC with state space S, states $s, s' \in S$, $G \subseteq S$ and time interval I:*

$$s \preceq s' \quad \textit{implies} \quad p^{\max}(s, I) \;\leq\; p^{\max}(s', I).$$

One may now be tempted to first minimise an IMC wrt. simulation preorder or its corresponding equivalence $\preceq \cap \preceq^{-1}$, but it turns out that checking a simulation relation between probabilistic models such as IMCs is computationally involved [1,67]. In the sequel, we will see that simulation preorders are nonetheless crucial to obtain more aggressive abstraction techniques for IMCs.

Interval abstraction. Compositional bisimulation minimisation has been applied to several examples yielding substantial state-space reductions. It allowed the analysis of IMCs (in fact, CTMCs) that could not be analysed without compositional minimisation [39,30,32]. With the advent of increasingly complex systems, more radical reduction techniques are needed. In the sequel, we present a recent framework to perform aggressive abstraction of IMCs in a compositional manner [49]. The key idea is to (again) partition the state space, but rather requiring that each partition solely consists of equivalent (strong or weak bisimilar) states, we are more liberal, and in fact allow for any state space partitioning. As a result, a state s is not bisimilar to its partition (as for bisimulation), but instead

its partition *simulates* s. Intuitively speaking, this means that all behaviour of s can be mimicked, but perhaps that the partition exhibits more behaviours than s. As the partition is aimed to be coarser than in the case of bisimulation, a central question is which measures are preserved, i.e., what does a maximal (time-bounded) reachability probability computed on the minimised IMC imply for the original IMC?

In the remainder of this section, we assume that IMCs are *uniform*.

Definition 11 (Uniform IMC). *An IMC is* uniform *if for any state s we have that $MT(s) \neq \emptyset$ implies $E(s) = \lambda$ for a given fixed $\lambda \in \mathbb{R}_{>0}$.*

The residence time in any state with at least one Markovian transition is thus governed by the same exponential distribution. Although this seems a rather severe restriction, there is an important class of systems for which this applies, viz. IMCs in which delays are imposed in a compositional manner using the constraint-oriented specification style. The point is that any CTMC can be transformed by a simple linear-time procedure into a weak bisimilar uniform CTMC [3]. Consider the specification $P \,||_A\, D_p$ where P is an IMC with only interactive transitions, i.e., P is an LTS, and D_p is a CTMC, probably enhanced with a start action α and end action β as explained before. The purpose of D_p is to impose a random delay between the occurrence of α and β in P. This is modeled as an arbitrary, finite-state CTMC. We can now transform D into its uniform counterpart $\widehat{D_p}$, say. As $D_p \approx \widehat{D_p}$ and $\approx$ is substitutive wrt. parallel composition, it follows that the non-uniform IMC $P \,||_A\, D_p$ is weak bisimilar to the uniform IMC $P \,||_A\, \widehat{D_p}$. (Several operators are preserving uniformity, see [38].)

Let IMC $\mathcal{I}$ be uniform. Our abstraction technique for $\mathcal{I}$ is a natural mixture of abstraction of labeled transition systems by *modal* transition systems [51,52] and abstraction of probabilities by *intervals* [27,48]. This combination yields *abstract* IMCs.

Definition 12 (Abstract IMC). *An* abstract *IMC is a tuple $\mathcal{I} = (S, Act, L, \mathbf{P}_l, \mathbf{P}_u, \lambda, s_0)$ with S, s_0 and Act as before, and*

- $L : S \times Act \times S \to \mathbb{B}_3$, *a* three-valued labeled *transition function*
- $\mathbf{P}_l, \mathbf{P}_u : S \times S \to [0,1]$, *lower/upper transition* probability bounds *s.t.*

$$\mathbf{P}_l(s,S) \leq 1 \leq \mathbf{P}_u(s,S) \quad \textit{and}$$

- $\lambda \in \mathbb{R}_{>0}$, *an exit rate.*

Here $\mathbb{B}_3 = \{\bot, ?, \top\}$ is the complete lattice with the ordering $\bot < ? < \top$ and meet ($\sqcap$) and join ($\sqcup$) operations. The labeling $L(s, \alpha, s')$ identifies the transition "type": $\top$ indicates a must-transition, ? a may-transition, and $\bot$ the absence of a transition. $\mathbf{P}_l(s,s')$ is the minimal one-step probability to move from s to s', whereas $\mathbf{P}_u(s,s')$ is the maximal one-step probability between these states. Given these bounds, the IMC can move from s to s' with any probability in the interval $[\mathbf{P}_l(s,s'), \mathbf{P}_u(s,s')]$. Any uniform IMC is an AIMC without may-transitions and for which $\mathbf{P}_l(s,s') = \mathbf{P}_u(s,s')$. The requirement $\mathbf{P}_l(s,S) \leq 1 \leq$

$\mathbf{P}_u(s, S)$ ensures that in any state s, a distribution μ over the direct successor states of s can be chosen such that for any s' we have: $\mathbf{P}_l(s, s') \leq \mu(s') \leq \mathbf{P}_u(s, s')$.

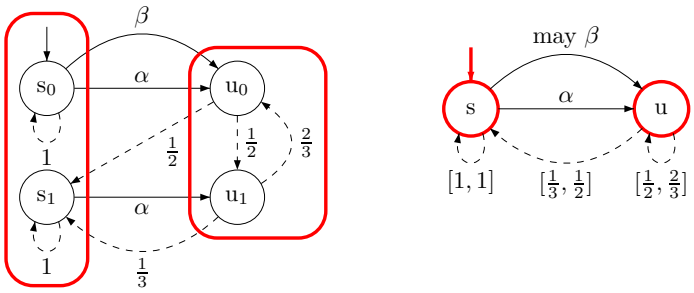
Let us now describe how to perform abstraction of an (A)IMC. As stated above, the principle is to partition the state space by grouping concrete states to abstract states. For concrete state space S and abstract state space S', let $\alpha : S \to S'$ map states to their corresponding abstract ones, i.e., $\alpha(s)$ denotes the abstract state of s and $\alpha^{-1}(s') = \gamma(s')$ is the set of concrete states that are mapped onto s'. α is called the abstraction function whereas $\gamma = \alpha^{-1}$ is known as the concretization function.

Definition 13 (Abstraction). *For an AIMC $\mathcal{I} = (S, Act, L, \mathbf{P}_l, \mathbf{P}_u, \lambda, s_0)$, the abstraction function $\alpha : S \to S'$ induces the AIMC $\alpha(\mathcal{I}) = (S', Act, L', \mathbf{P}'_l, \mathbf{P}'_u, \lambda, \alpha(s_0))$, where:*

- $L'(s', \beta, u') = \begin{cases} \top & \text{if } \bigsqcup_{u \in \gamma(u')} L(s, \beta, u) = \top \text{ for all } s \in \gamma(s') \\ \bot & \text{if } \bigsqcup_{u \in \gamma(u')} L(s, \beta, u) = \bot \text{ for all } s \in \gamma(s') \\ ? & \text{otherwise} \end{cases}$
- $\mathbf{P}'_l(s', u') = \min_{s \in \gamma(s')} \sum_{u \in \gamma(u')} \mathbf{P}_l(s, u)$
- $\mathbf{P}'_u(s', u') = \min(1, \max_{s \in \gamma(s')} \sum_{u \in \gamma(u')} \mathbf{P}_u(s, u))$

There is a must-transition $s' \xrightarrow{\alpha} u'$ if any concrete version $s \in \gamma(s')$ exhibits such transition to some state in $\gamma(u)$. There is no transition between s' and u' if there is no such transition from $s \in \gamma(s')$ to $\gamma(u)$. In all other cases, we obtain a may-transition $s' \xrightarrow{\alpha} u'$.

Example 4. Consider the uniform IMC depicted in the figure below on the left, and ket $S' = \{s, u\}$ be the abstract state space. Assume the abstraction is defined by $\alpha(u_0) = \alpha(u_1) = u$, and $\alpha(s_0) = \alpha(s_1) = s$. This yields the abstract IMC depicted on the right. As $s_0 \xrightarrow{\alpha} u_0$ and $s_1 \xrightarrow{\alpha} u_1$, there is a must-transition labeled by α from s to u. Although $s_0 \xrightarrow{\beta} u_0$, s_1 has no β-transition to u_0 or u_1. Accordingly, we obtain a may-transition labeled with β between s and u. As $\mathbf{P}(u_0, s_1) = \frac{1}{2}$ and $\mathbf{P}(u_1, s_1) = \frac{1}{3}$, we obtain that $\mathbf{P}_l(u, s) = \frac{1}{3}$ and $\mathbf{P}_u(u, s) = \frac{1}{2}$. The other probability intervals are justified in a similar way.



The formal relationship between an AIMC and its abstraction is given by a simulation relation which is in fact a combination of probabilistic simulation on

IMCs as defined before (with a slight adaptation to deal with intervals) and the concept of refinement on modal transition systems [52]. Let $\mathbf{T}(s)$ denote the set of probability distributions that exist in state s and that satisfy all bounds of the probability intervals of the outgoing Markovian interval transitions of s.

Definition 14 (Strong simulation). *For AIMC* $\mathcal{I} = (S, Act, L, \mathbf{P}_l, \mathbf{P}_u, \lambda, s_0)$, $R \subseteq S \times S$ *is a* simulation relation, *iff for any* $(s, s') \in R$ *it holds:*

1a. for all $\alpha \in Act$ *and* $u \in S$ *with* $L(s, \alpha, u) \neq \bot$ *there exists* $u' \in S$ *with* $L(s', \alpha, u') \neq \bot$ *and* $(u, u') \in R$,
1b. for all $\alpha \in Act$ *and* $u' \in S$ *with* $L(s', \alpha, u') = \top$ *there exists* $u \in S$ *with* $L(s, \alpha, u) = \top$ *and* $(u, u') \in R$, *and*
2. $L(s, \tau, u) \neq \top$ *for all* $u \in S$, *implies for all* $\mu \in \mathbf{T}(s)$ *there exists* $\mu' \in \mathbf{T}(s')$ *and* $\Delta : S \times S \to [0, 1]$ *such that for all* $u, u' \in S$*:*

(a) $\Delta(u, u') > 0 \implies uRu'$ *(b)* $\Delta(u, S) = \mu(u)$ *(c)* $\Delta(S, u') = \mu'(u')$

We write $s \preceq s'$ if $(s, s') \in R$ for some simulation R and $\mathcal{I} \preceq \mathcal{I}'$ for AIMCs $\mathcal{I}$ and $\mathcal{I}'$ with initial states s_0 and s_0', if $s_0 \preceq s_0'$ in the disjoint union of $\mathcal{I}$ and $\mathcal{I}'$.

Let us briefly explain this definition. Item 1a requires that any may- or must-transition of s must be reflected in s'. Item 1b requires that any must-transition of s' must match some must-transition of s, i.e., all required behavior of s' stems from s. Note that this allows a must-transition of s to be mimicked by a may-transition of s'. Condition (2) is the same as in the defininition of simulation for IMCs, except that the set of distributions in a state in an IMC is a singleton, whereas for AIMCs this set can be infinite.

Theorem 12. *[49] For any AIMC* $\mathcal{I}$ *and abstraction function* α, $\mathcal{I} \preceq \alpha(\mathcal{I})$.

As this abstraction is coarser than bisimulation, a significantly larger state-space reduction may be achieved and peak memory consumption is even further reduced. The notion of parallel composition and hiding, as defined for IMCs can now be lifted to AIMCs in a natural manner, and it can be shown that

Theorem 13. *[49]* $\preceq$ *is a pre-congruence wrt. parallel composition and hiding.*

This result provides us the means to carry out abstraction on (A)IMCs in a fully compositional manner. Suppose the system-to-be-analysed is of the form

$$\mathcal{I} \;=\; \mathcal{I}_1 \,||_{A_1}\, \mathcal{I}_2 \,||_{A_2}\, \ldots \,||_{A_N}\, \mathcal{I}_N,$$

i.e., a parallel composition of N IMCs. Let $\alpha(\mathcal{I}_i)$ be the abstraction of IMC $\mathcal{I}_i$, for $0 < i \leq N$. Thanks to the property that strong simulation is substitutive wrt. parallel composition, it follows from the fact that $\mathcal{I}_i \preceq \alpha(\mathcal{I}_i)$, for $0 < i \leq N$, that:

$$\mathcal{I}_1 \,||_{A_1}\, \mathcal{I}_2 \,||_{A_2}\, \ldots \,||_{A_N}\, \mathcal{I}_N \;\preceq\; \alpha(\mathcal{I}_1) \,||_{A_1}\, \alpha(\mathcal{I}_2) \,||_{A_2}\, \ldots \,||_{A_N}\, \alpha(\mathcal{I}_N).$$

5 IMCs as semantical model

Much of computer science is about specification formalisms. Domain specific languages as well as universal notations are being promoted by various interest groups and taken up by standardization bodies. Some of them appeal due to their graphical notation convenience, such as the UML, others appeal because they clarify the aspects of a certain domain. One example of the latter is AADL, the Architectural Analysis and Design Language [28]. For many of these languages the work is considered done once the syntax is fixed, and an intuitive explanation of the semantics is provided. Formalizing these intuitions is sometimes a task for legions of scientists: The conception of Statecharts for instance has lead to several dozens of different semantics, and more are on the horizon. Still, one of the lessons generally learnt from these experiences is that a good semantics is compositional [29], a semantics that provides a meaning to an object based on a composition of the semantics of its parts. If the composition adheres to simple-to-grasp rules, this semantics can become consensus. Compositionality is a fundamental and highly desirable property of a semantics: it enables compositional reasoning, i.e. analyzing complex systems by breaking them down into their constituting parts. Examples *par excellence* of simple-to-grasp rules have been given before: parallel composition and hiding.

A clean and well-understood semantics is a necessity for model-based evaluation of such languages. It is as simple as that. Whenever performance figures or correctness claims are presented for UML fragments or the like, they are specific to the semantics chosen, and in case that semantics is neither commonly agreed nor easy-to-grasp, doubts remain concerning the general validity of such claims.

Dynamic fault trees. Let us consider a classical domain specific language, known as fault trees. Fault trees were first planted in the youth of civil nuclear energy, as means to systematically quantify the risk of a catastrophic hazard [63] in a plant. A fault tree is a diagrammatical variation of a boolean function, drawn in a tree-structured manner where the leaves correspond to boolean variables. These leaves represent basic operational units of the plant such as valves and pipes. The failure of an operational components flips the corresponding boolean value to true. If the entire function evaluates to true, a catastrophic event is supposed to be unavoidable. Fault trees have been standardised, and their use is prescribed in many engineering areas. A classical fault tree is static, the order of failure occurences is assumed not important, and components cannot be replaced dynamically by spare components. If considering such extensions, one arrives at the diagrammatical notation of dynamic fault trees (DFT) [25].

The semantics of a dynamic fault tree can no longer be mapped directly on a boolean function, but instead needs a state-transition graph representation to reflect the system dynamics. If one assumes that failure occurences follow exponential laws, which is a standard and sometimes justified assumption, it seems natural to expect that the resulting model is a CTMC. Actually, the first complete formalisation attempted [21] aimed at providing a CTMC semantics, but revealed a number of ambiguities in the DFT framework. Most notably, in some

instances of DFTs non-determinism arises. This is where IMC and its compositionality property can play a pivotal role: The work of Crouzen *et al.* [12,11,10] provides a clean and elegant compositional semantics, a semantics that maps on IMC. More precisely, the semantics takes up ideas of I/O-automata [53], and uses *input/output* interactive Markov chains (I/O-IMC). I/O-IMC are restricted versions of IMC that allow for non-blocking communication. The semantics is fully compositional: The semantics of each DFT element is an I/O-IMC. The semantics of a DFT is then obtained by parallel composing the I/O-IMC semantics of all its elements.

Example 5. As an example, we demonstrate this approach for a SPARE gate, a functional unit that makes a redundant unit of functionality available (the spare), in case the original unit (the primary) fails [12]. Figure 3 shows the I/O-IMC semantics of a DFT consisting of a SPARE gate A having a primary B and a spare C.

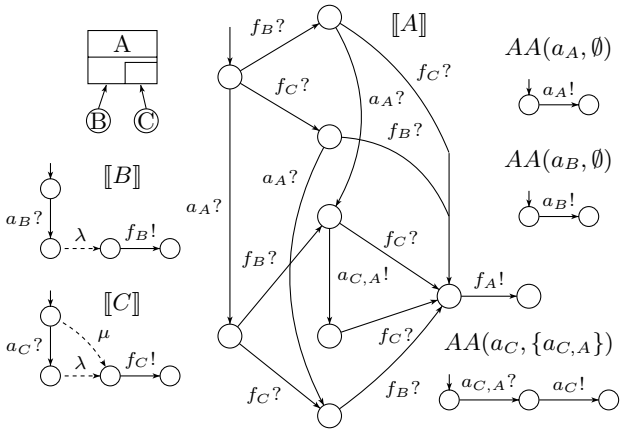


Fig. 3. A DFT example and six I/O-IMCs that model its behavior [12].

The I/O-IMC of the DFT is obtained by parallel composing the six IMCs, properly synchronised. The congruence property established before is inherited by I/O-IMCs and enables compositional aggregation to combat the state-space explosion problem existing in DFTs, see [10,12].

Archtectural Description Languages. Hardware/software (HW/SW) co-design of safety-critical embedded systems such as on-board systems that appear in the aerospace domain is a very complex and highly challenging task. Component-based design is an important paradigm here to master this design complexity while, in addition, allowing for reusability. As safety-critical systems are subject to hardware and software faults, the adequate modeling of faults, their likelihood of occurrence, and the way in which a system can recover from faults,

are essential to a model-based approach for safety-critical systems. To overcome these shortcomings one needs an enriched practical component-based modeling approach with appropriate means for modeling probabilistic fault behavior.
To warrant acceptance by design engineers in, e.g., aerospace industry and the automotive engineers, efforts have been spent to based on the Architecture Analysis and Design Language (AADL) [28], a design formalism that is standardised by the Society of Automotive Engineers. Among these efforts, Arcade [9] has adopted the DFT work mentioned above to the recent AADL Error Model Annex, and provides a map of each of the components on an I/O-IMC, again in a fully compositional manner. This is in spirit similar to the work performed in the ESA project COMPASS [13,14,15], where IMC are targetted to model nominal and probabilistic fault behaviour, fault propagation and recovery, and degraded modes of operation. The integration of nominal behavior and error models basically boils down to a parallel composition of a variable-decorated transition system (which is a semantically an IMC) and an IMC.

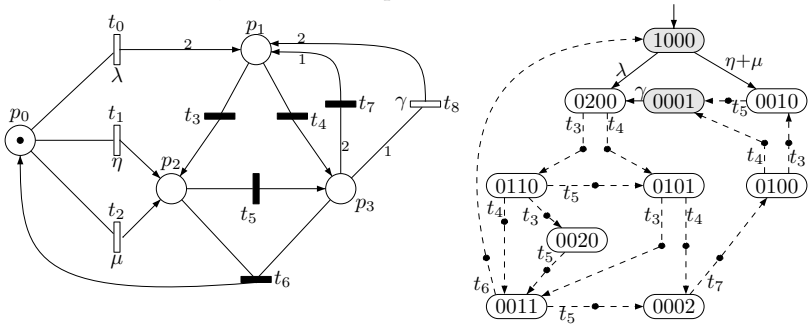
Generalised stochastic Petri nets. Generalised Stochastic Petri Nets (GSPNs) are a well-established modelling formalism for performance and dependability evaluation, supporting stochastic timed behavior and weighted immediate choices [54,55]. To this end, timed transitions and immediate transitions are supported in a GSPN. Performance evaluation of a GSPN proceeds at the level of the reachability (or: marking) graph. That graph is transformed into a CTMC, for which efficient steady-state and transient solvers are at hand. This evaluation trajectory was pioneered by the tool GreatSPN [20], nowadays it is implemented in a plethora of tools.

However, it is notoriously overlooked that the above evaluation trajectory is incomplete. It is restricted to *confusion-free* GSPNs. Confusion arises if a firing sequence admits the simultaneous enabling of multiple non-conflicting immediate transitions. GSPNs equip immediate transitions with global priority levels and globally assigned weights to diminish the occurrence of such nondeterministic choices. But priorities and weights do not, and cannot, eliminate confusion in its full entirety. The presence of nondeterminism, however, makes it impossible to associate an unambiguous stochastic process to such nets.

Recently we managed to attack this principal problem [40]. We have taken up earlier thoughts on nondeterministic GSPN semantics [37] to come up with an IMC semantics for GSPNs. Actually, this semantics is not more than a reinterpretation of the marking graph as an IMC. With the analysis results reported in this paper, this means that also confused GSPNs can now be analysed. This was not possible before.

Example 6. Consider the GSPN depicted in the figure below on the left where solid bars depict immediate transitions and open bars represent delayed transitions. This GSPN is confused. In marking $(0, 0, 1, 1)$, for instance, the set of reachable tangible markings is $\{(1, 0, 0, 0), (0, 0, 0, 1)\}$. If the enabled transition t_5 is chosen, the tangible marking $(0, 0, 0, 1)$ is reached almost surely. However, if enabled transition t_6 is chosen, we enter the tangible marking $(1, 0, 0, 0)$ almost

surely. Hence, the next tangible markings depends on the way the nondeterminism in $(0,0,1,1)$ is resolved and cannot be quantified. The usual way to deal with this situation is to equip transitions t_5 and t_6 with weigths. The marking graph of the GSN is depicted in the figure on the right. Here, solid arrows depict Markovian transitions, and dashed arrows correspond to the firing of immediate transitions in the net, and are interpreted as τ-labeled IMC transitions.



Statecharts. A modelling environment used by engineers in several avionic and automotive companies like AIRBUS or BMW is STATEMATE, a Statechart-based tool-set. To enable performance and dependability evaluation of Statemate designs, the German special research initiative AVACS has spent considerable energy to a connection between Statemate and IMC [8,38,65,42].

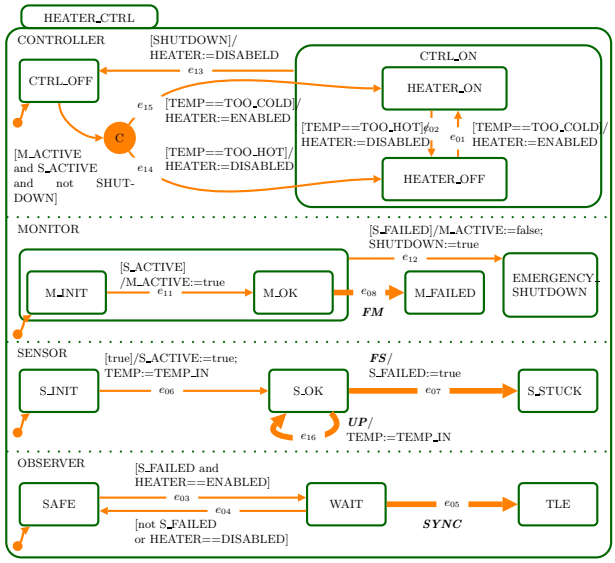


Fig. 4. A Statechart [8]

One key feature of this approach is that the model construction steps rely heavily on compositional properties of the IMC model, and employ precisely the

constraint-oriented specification style advocated in Section 2. For this, the design comprises distinguished *delay transitions*. In the figure, these are FM, failure of monitor and FS, failure of sensor). These transitions have an effect or are affected by the advance of time. Mostly, delay transitions indicate component failures, but the concept is more flexible. *Delay distributions*, in the form of continuous probability distributions affecting the occurrence of the delay transitions are incorporated via the *elapse*-operator. Symbolic (i. e. BDD-based) representations and compositional methods are exploited to keep the model sizes manageable.
The complexity challenges posed by this problem domain could only be addressed by (1) performing a state space reduction on the non-deterministic part of the model by means of a *symbolic* minimisation capable of handling huge state spaces and (2) constraint-oriented specification of time-constraints *after* this reduction into the model. The developed technology was applied to a non-trivial case study from the train control domain with an explication of the improvements contributed by each of the relevant translation steps. In Table 5 we quote a fraction of the relevant information from [8], illustrating the effects and costs of compositional minimisation.

Tracks –	Compositional Construction			Final Quotient IMC	
Choices	States	Transitions	Time (sec.)	States	Transitions
1 – 1	71	261	18.70	16	55
1 – 2	79	325	22.58	29	101
1 – 3	99	419	26.87	39	143
1 – 4	119	513	30.43	49	185
2 – 1	731	3888	31.72	187	1006
2 – 2	1755	11059	39.55	665	3442
2 – 3	3127	20737	47.71	1281	6991
2 – 4	4899	33495	57.83	2097	11780
3 – 1	10075	75377	50.01	2573	18260
3 – 2	53858	387501	293.53	16602	112011
3 – 3	134555	1061958	1114.82	44880	320504
4 – 1	143641	1343747	785.30	35637	313270
4 – 2	1350908	11619969	243687.33	416274	3452502

Fig. 5. Composition and minimisation statistics [8]

6 Concluding remarks

This paper has presented an overview of foundational, algorithmic and pragmatic aspects of IMCs, a simple generalisation of both CTMCs and LTS with a fully compositional semantics. There are other approaches that give a compositional semantics in a continuous time Markov setting, among them popular formalisms

such as PEPA [43], EMPA [6] or MTIPP [41], the latter being the semantic basis of the PRISM toolkit [44] in 'ctmc' mode. None of these formalisms has the properties that IMCs possess. In particular, they do not extend classical concurrency models in a conservative fashion. For each of these calculi the role of an atomic action is particular. This affects the synchronisation of actions, and thus the final performance results – in different ways for each of these calculi. It is not easy to explain what is happening precisely, and this is not the topic of this paper; the interested reader may consult [17]. In IMC, the separation of delays and actions allows to treat action synchronisation as in standard concurrency models. It is surprising that given the advantages of IMCs, recent approaches for CTMC-variants of process calculi for mobility and service-oriented computing [59] and interesting new developments in structured operational semantics for such calculi [50,24] do not adopt this approach.

An extension of IMC towards time-inhomogeneous continuous-time dynamics is provided in [33]. In the discrete time setting, a model class with similarly distinguishing properties is provided by probabilistic automata [60]. Probabilistic automata can be integrated into the IMC model, retaining full compositionality [26].

We have reviewed the theoretical basis of IMC, and have discussed two recent algorithmic achievements that foster the applicability of IMC: analysis techniques in the presence of nondeterminism, and compositional abstraction techniques. IMCs practical relevance has been highlighted by reviewing in applications of various domains, ranging from dynamic fault trees to generalized stochastic Petri nets. While the first generation of tool support, CADP, has found several academic and non academic uses, the recent algorithmic advances described in this paper are not yet fully integrated in a tool. This is a major topic of ongoing work.

References

1. Baier, C., Engelen, B., Majster-Cederbaum, M. E.: Deciding bisimilarity and similarity for probabilistic processes. *Journal of Computer and System Sciences* **60** (2000) 187–231
2. Baier, C., Haverkort, B. R., Hermanns, H., Katoen, J.-P.: Model-checking algorithms for continuous-time Markov chains. *IEEE TSE* **29** (2003) 524–541
3. Baier, C., Katoen, J.-P., Hermanns, H., Wolf, V.: Comparative branching-time semantics for Markov chains. *Information and Computation* **200** (2005) 149–214
4. Barendregt, H.: The quest for correctness. In: *Images of SMC Research 1996*. Stichting Mathematisch Centrum (1996) 39–58
5. Bergstra, J. A., Ponse, A., (editors), S. A. S.: *Handbook of Process Algebra*. Elsevier Publishers B.V. (2001)
6. Bernardo, M., Gorrieri, R.: Corrigendum to "A tutorial on EMPA: A theory of concurrent processes with nondeterminism, priorities, probabilities and time" - [TCS 202 (1998) 1–54]. *Theoretical Computer Science* **254** (2001) 691–694
7. Bertsekas, D.: *Dynamic Programming and Optimal Control*. Vol. II. Athena Scientific (1995)

8. Böde, E., Herbstritt, M., Hermanns, H., Johr, S., Peikenkamp, T., Pulungan, R., Rakow, J., Wimmer, R., Becker, B.: Compositional dependability evaluation for STATEMATE. *IEEE TSE* **35** (2009) 274–292
9. Boudali, H., Crouzen, P., Haverkort, B. R., Kuntz, M., Stoelinga, M. I. A.: Architectural dependability evaluation with Arcade. In: *Dependable Systems and Networks (DSN)*. IEEE (2008) 512–521
10. Boudali, H., Crouzen, P., Stoelinga, M.: A compositional semantics for dynamic fault trees in terms of interactive Markov chains. In: *Automated Technology for Verification and Analysis (ATVA). LNCS*, Vol. 4762. Springer Verlag (2007) 441–456
11. Boudali, H., Crouzen, P., Stoelinga, M. I. A.: Dynamic fault tree analysis using input/output interactive Markov chains. In: *Dependable Systems and Networks (DSN)*. IEEE (2007)
12. Boudali, H., Crouzen, P., Stoelinga, M. I. A.: Rigorous, compositional, and extensible framework for dynamic fault tree analysis. *IEEE Transactions on Secure and Dependable Computing* **7** (2009) 128–143
13. Bozzano, M., Cimatti, A., Katoen, J.-P., Nguyen, V., Noll, T., Roveri, M.: Codesign of dependable systems: A component-based modelling language. In: *Proc. 7th Int. Conf. on Formal Methods and Models for Co-Design (MEMOCODE 2009)*. IEEE CS Press (2009) 121–130
14. Bozzano, M., Cimatti, A., Katoen, J.-P., Nguyen, V., Noll, T., Roveri, M.: The COMPASS approach: Correctness, modelling and performability of aerospace systems. In: *Proc. 28th Int. Conf. on Computer Safety, Reliability and Security (SAFECOMP 2009). Lecture Notes in Computer Science*, Vol. 5775. Springer (2009) 173–186
15. Bozzano, M., Cimatti, A., Katoen, J.-P., Nguyen, V., Noll, T., Roveri, M.: Safety, dependability and performance analysis of extended AADL models. *The Computer Journal* (2010)
16. Bravetti, M., Hermanns, H., Katoen, J.-P.: YMCA: Why Markov chain algebra? In: *Proceedings of the Workshop Essays on Algebraic Process Calculi. Electronic Notes in Theoretical Computer Science*, Vol. 162. Elsevier (2006) 107–112
17. Brinksma, E., Hermanns, H.: Process algebra and Markov chains. In: *Lectures on Formal Methods and Performance Analysis (FMPA). LNCS*, Vol. 2090. Springer (2001) 183–231
18. Buchholz, P.: Exact and ordinary lumpability in finite markov chains. *J. of Applied Probability* **31** (1994) 59–75
19. Chehaibar, G., Zidouni, M., Mateescu, R.: Modeling multiprocessor cache protocol impact on MPI performance. In: *IEEE International Workshop on Quantitative Evaluation of Large-Scale Systems and Technologies*. IEEE (2009)
20. Chiola, G., Franceschinis, G., Gaeta, R., Ribaudo, M.: GreatSPN 1.7: Graphical editor and analyzer for timed and stochastic Petri nets. *Performance Evaluation* **24** (1995) 47–68
21. Coppit, D., Sullivan, K. J., Dugan, J. B.: Formal semantics for computational engineering: A case study on dynamic fault trees. In: *ISSRE*. IEEE Computer Society (2000) 270–282
22. Coste, N., Garavel, H., Hermanns, H., Hersemeule, R., Thonnart, Y., Zidouni, M.: Quantitative evaluation in embedded system design: Validation of multiprocessor multithreaded architectures. In: *Design, Automation and Test in Europe (DATE)*. IEEE (2008) 88–89

23. Coste, N., Hermanns, H., Lantreibecq, E., Serwe, W.: Towards performance prediction of compositional models in industrial GALS designs. In: *Computer Aided Verification (CAV)*. Vol. 5643. Springer (2009) 204–218
24. De Nicola, R., Latella, D., Loreti, M., Massink, M.: Rate-based transition systems for stochastic process calculi. In: *Int. Colloquium on Automata, Languages and Programming (ICALP)*. *LNCS*, Vol. 5556. Springer (2009) 435–446
25. Dugan, J., Bavuso, S., Boyd, M.: Dynamic fault-tree models for fault-tolerant computer systems. *IEEE Transactions on Reliability* **41** (1992) 363–77
26. Eisentraut, C., Hermanns, H., Zhang, L.: On probabilistic automata in continuous time. In: *IEEE Symposium on Logic in Computer Science (LICS)*. IEEE (2010)
27. Fecher, H., Leucker, M., Wolf, V.: Don't know in probabilistic systems. In: *Software Verification*. *LNCS*, Vol. 3925. Springer-Verlag (2006) 71–88
28. Feiler, P. H., Rugina, A.: Dependability modeling with the Architecture Analysis & Design Language (AADL). Technical Note CMU/SEI-2007-TN-043, CMU Software Engineering Institute (2007)
29. Frenkel, K. A., Milner, R.: An interview with Robin Milner. *CACM* **36** (1993) 90–97
30. Garavel, H., Hermanns, H.: On combining functional verification and performance evaluation using CADP. In: *Formal Methods Europe (FME)*. *LNCS*, Vol. 2391. Springer-Verlag (2002) 410–429
31. Garavel, H., Mateescu, R., Lang, F., Serwe, W.: CADP 2006: A toolbox for the construction and analysis of distributed processes. In: *Computer-Aided Verification (CAV)*. *LNCS*, Vol. 4590. Springer-Verlag (2007) 158–163
32. Gilmore, S., Hillston, J., Ribaudo, M.: An efficient algorithm for aggregating PEPA models. *IEEE Trans. Software Eng.* **27** (2001) 449–464
33. Han, T., Katoen, J.-P., Mereacre, A.: Compositional modeling and minimization of time-inhomogeneous Markov chains. In: *Hybrid Systems: Computation and Control (HSCC)*. *LNCS*, Vol. 4981. Springer-Verlag (2008) 244–258
34. Haverkort, B. R.: *Performance of Computer Communication Systems: A Model-Based Approach*. John Wiley & Sons (1998)
35. Hermanns, H.: *Interactive Markov Chains and the Quest for Quantified Quality*. *LNCS*, Vol. 2428. Springer (2002)
36. Hermanns, H., Herzog, U., Katoen, J.-P.: Process algebra for performance evaluation. *Theoretical Computer Science* **274** (2002) 43–87
37. Hermanns, H., Herzog, U., Mertsiotakis, V., Rettelbach, M.: Exploiting stochastic process algebra achievements for generalized stochastic Petri nets. In: *Petri Nets and Performance Models (PNPM)*. IEEE (1997) 183–192
38. Hermanns, H., Johr, S.: Uniformity by construction in the analysis of nondeterministic stochastic systems. In: *Dependable Systems and Networks (DSN)*. IEEE (2007) 718–728
39. Hermanns, H., Katoen, J.-P.: Automated compositional Markov chain generation for a plain-old telephone system. *Science of Comp. Progr.* **36** (2000) 97–127
40. Hermanns, H., Katoen, J.-P., Neuhäußer, M. R., Zhang, L.: GSPN model checking despite confusion. Technical report, RWTH Aachen University (2010)
41. Hermanns, H., Rettelbach, M.: Syntax, Semantics, Equivalences, and Axioms for MTIPP. In Herzog, U., Rettelbach, M. (eds.): *Proc. of the 2nd Int. Workshop on Process Algebras and Performance Modelling*. *Arbeitsberichte des IMMD*, Vol. 27(4). Universität Erlangen (1994)
42. Hermanns, H., Johr, S.: May we reach it? or must we? in what time? with what probability? In: *Measurement, Modelling and Evaluation of Computer and Communication Systems (MMB)*. VDE Verlag (2008) 125–140

43. Hillston, J.: *A Compositional Approach to Performance Modelling*. Cambridge University Press (1996)
44. Hinton, A., Kwiatkowska, M. Z., Norman, G., Parker, D.: PRISM: A tool for automatic verification of probabilistic systems. In: *Tools and Algorithms for the Construction and Analysis of Systems (TACAS). LNCS*, Vol. 3920. Springer (2006) 441–444
45. Hoare, C., Brookes, S., Roscoe, A.: A theory of communicating sequential processes. *J. ACM* **31** (1984) 560–599
46. Jonsson, B.: Simulations between specifi
cations of distributed systems. In: *Concurrency Thepry (CONCUR). LNCS*, Vol. 527. Springer (1991) 346–360
47. Katoen, J.-P., Kemna, T., Zapreev, I. S., Jansen, D. N.: Bisimulation minimisation mostly speeds up probabilistic model checking. In: *Tools and Algorithms for the Construction and Analysis of Systems (TACAS). LNCS*, Vol. 4424. Springer-Verlag (2007) 87–102
48. Katoen, J.-P., Klink, D., Leucker, M., Wolf, V.: Three-valued abstraction for continuous-time Markov chains. In: *Computer Aided Verification (CAV). LNCS*, Vol. 4590. Springer (2007)
49. Katoen, J.-P., Klink, D., Neuhäußer, M. R.: Compositional abstraction for stochastic systems. In: *Formal Modeling and Analysis of Timed Systems (FORMATS). LNCS*, Vol. 5813. Springer (2009) 195–211
50. Klin, B., Sassone, V.: Structural operational semantics for stochastic process calculi. In: *Foundations of Software Science and Computational Structures (FoSSaCS). LNCS*, Vol. 4962. Springer (2008) 428–442
51. Larsen, K. G.: Modal specifications. In: *Automatic Verification Methods for Finite State Systems. LNCS*, Vol. 407. Springer-Verlag (1989) 232–246
52. Larsen, K. G., Thomsen, B.: A modal process logic. In: *IEEE Symposium on Logic in Computer Science (LICS)*. IEEE (1988) 203–210
53. Lynch, N. A., Tuttle, M. R.: An introduction to input/output automata. *CWI Quarterly* **2** (1989) 219–246
54. Marsan, M. A., Balbo, G., Chiola, G., Conte, G., Donatelli, S., Franceschinis, G.: An introduction to generalized stochastic Petri nets. *Microelectronics and Reliability* **31** (1991) 699–725
55. Marsan, M. A., Balbo, G., Conte, G., Donatelli, S., Franceschinis, G.: *Modelling with Generalized Stochastic Petri Nets*. John Wiley & Sons (1995)
56. Neuhäußer, M. R., Katoen, J.-P.: Bisimulation and logical preservation for continuous-time Markov decision processes. In: *Concurrency Theory (CONCUR). LNCS*, Vol. 4703. Springer (2007) 412–427
57. Neuhäußer, M. R., Stoelinga, M., Katoen, J.-P.: Delayed nondeterminism in continuous-time Markov decision processes. In: *Foundations of Software Science and Computational Structures (FoSSaCS). LNCS*, Vol. 5504. Springer (2009) 364–379
58. Neuhäußer, M. R.: *Model Checking Nondeterministic and Randomly Timed Systems*. PhD thesis, RWTH Aachen University / University of Twente (2010)
59. Prandi, D., Quaglia, P.: Stochastic COWS. In: *Int. Conf. on Service-Oriented Computing (ICSOC). LNCS*, Vol. 4749. Springer (2009) 245–256
60. Segala, R.: *Modeling and Verification of Randomized Distributed Real-Time Systems*. PhD thesis, Laboratory for Computer Science, Massachusetts Institute of Technology (1995)
61. URL: (http://portal.acm.org/citation.cfm?id=1451820)

62. van Glabbeek, R. J., Weijland, W. P.: Branching time and abstraction in bisimulation semantics. *J. ACM* **43** (1996) 555–600
63. Veseley, W., Goldberg, F., Roberts, N., Haasl, D.: *Fault Tree Handbook*. US Nuclear Regulatory Commission, NUREG- 0492 (1981)
64. Vissers, C., Scollo, G., van Sinderen, M., Brinksma, E.: On the use of specification styles in the design of distributed systems. *Theor. Comput. Sci.* **89** (1991) 179–206
65. Wimmer, R., Herbstritt, M., Hermanns, H., Strampp, K., Becker, B.: Sigref – a symbolic bisimulation tool box. In: *Int. Symp. on Automated Technology for Verification and Analysis (ATVA). LNCS*, Vol. 4218. Springer-Verlag (2006) 477–492
66. Zhang, L., Neuhäußer, M. R.: Model checking interactive Markov chains. In: *Tools and Algorithms for the Construction and Analysis of Systems (TACAS). LNCS*, Vol. 6015. Springer (2010) 53–68
67. Zhang, L., Hermanns, H., Eisenbrand, F., Jansen, D. N.: Flow faster: Efficient decision algorithms for probabilistic simulations. *Logical Methods in Computer Science* **4** (2008)