Towards analysing nondeterministic specifications of stochastic systems

Holger Hermanns[1,2] Sven Johr[1]

[1]Department of Computer Science,
Saarland University, D-66123 Saarbrücken, Germany

[2]Faculty of Electrical Engineering, Mathematics and Computer Science,
University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands

Abstract. A continuous-time Markov decision process (CTMDP) is a generalisation of a continuous-time Markov chain in which both probabilistic and nondeterministic choices coexist. CTMDPs can intuitively be viewed as a common semantic model for confused generalised stochastic Petri nets, stochastic activity networks which are not well-defined, and for various stochastic process algebras including interactive Markov chains. Such performance and dependability models have so far resisted any kind of analysis, as a result of the presence of nondeterminism. In this paper we make the intuitive connection to CTMDPs more precise and show how interactive Markov chains can be mapped onto CTMDPs by a sequence of sound model transformations. Variants of bisimulation and trace equivalence are used to formally establish soundness aspects of the transformation. We apply the transformation to a workstation cluster example.

1 Introduction

Performance and dependability models are often developed using Markov-based modelling formalisms such as generalised stochastic Petri nets (GSPNs) [1], Markovian stochastic activity networks (SANs) [2] or Markovian stochastic process algebras [3–6]. Among these formalisms the presence or absence of nondeterminism is a subtle issue. Some formalisms (such as $\mathrm{EMPA_{gr}}$ [4]) are restricted to model systems without nondeterminism, or replace it by probabilism (such as PEPA). Others, including GSPNs [1], SANs [2], interactive Markov chains (IMCs) [6] and TIPP process algebra [5] instead incorporate nondeterministic features. To illustrate this phenomenon, Figure 1 displays a GSPN of a dependability model which has appeared in the literature [7,8], but whose behaviour does not describe a Markov chain, because of the presence of nondeterminism.

So far, the analysis of models developed in these and related formalisms was restricted to the subset that corresponds to continuous-time Markov chains (CTMCs), usually referred to as 'non-confused', 'well-defined', or 'well-specified' models [6,9–11]. All these notions are semantic notions, usually checked by an exhaustive exploration of the state space, which attempts to associate a unique CTMC with the specification. If this fails, the model is discarded and no analysis is carried out. In other words, no specification-level check is available, and the offered analysis algorithms are actually partial algorithms. Furthermore the precise answer what type of process to associate with 'confused', 'ill-defined' or 'ill-specified' models has not been addressed.

Continuous-time Markov decision processes (CTMDPs) [12–15] are generalisations of CTMCs in which both probabilistic and nondeterministic choices coexist. CTMDPs occur in many contexts, ranging from stochastic control theory [13] and stochastic schedul-

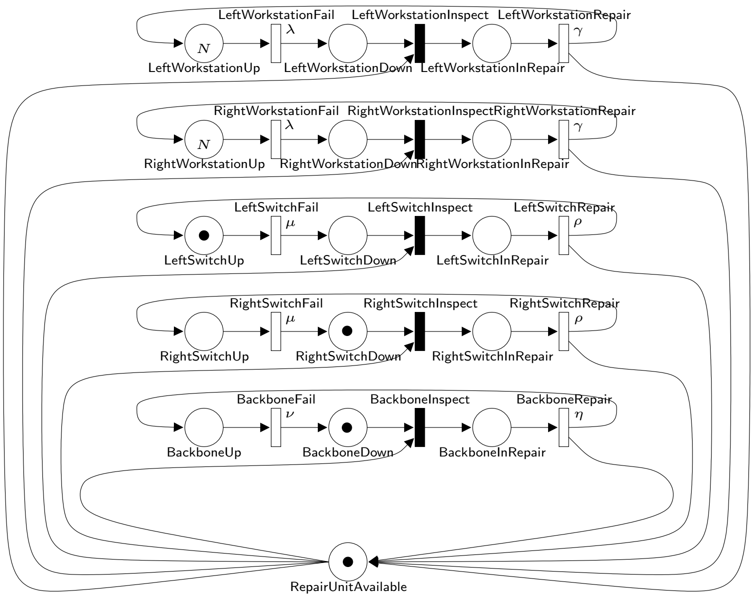

Fig. 1. A GSPN model of a repairable workstation cluster. Solid bars indicate immediate transitions, non-solid bars indicate timed transitions. In the depicted marking, the transitions BackboneInspect and RightSwitchInspect are in conflict. No probability or priority scheme is available to determine which of the transitions will fire next. The behaviour of this GSPN is thus partially nondeterministic.

ing [16, 17] to dynamic power management [18]. Analysis of such models is more expensive than for CTMCs [14, 19, 20].

Intuitively, the coexistence of nondeterminism and probabilism makes CTMDP appear as an appropriately general model for SANs, GSPNs, TIPP, or IMC models. This paper makes this intuition more precise. It shows how an interactive Markov chain can be transformed into a CTMDP by means of a sequence of model transformations. Each of these transformations is shown to preserve strong bisimulation, which serves as a correctness criterion for the entire transformation.

The IMC model is a combination of a nondeterministic labelled transition system and continuous-time Markov chains. It is the semantic model of the *calculus of interactive Markov chains* [6]. Furthermore, restricted versions of SANs and GSPNs have a semantics in terms of the IMC model as well, the restriction being that immediate probabilistic branching is excluded. As a result, our transformation can be used not only for mapping IMCs, but also for mapping restrictions of SANs and GSPNs onto CTMDPs.

Organisation of the paper. In Section 2 we discuss compositionality issues and give the required background on interactive Markov chains and continuous-time Markov decision processes. Section 3 describes the complete transformation process in great detail and contains results on the preservation properties. In Section 4 we give some details concerning an on-the-fly implementation of the transformation and present results of its application to the fault-tolerant workstation cluster mentioned above. Finally, Section 5 discusses some features of the transformation and concludes.

2 Context

This section describes the models and modelling formalisms that will be used throughout this paper. In this paper we assume all sets to be finite unless otherwise stated.

2.1 Open vs. closed system view

In the sequel we will tackle the problem how to transform an IMC into a CTMDP. The proposed transformation preserves important properties of the IMC, but it will *not* be a *compositional transformation. Compositionality* is often considered desirable, but a compositional transformation implicitly requires a compositional theory for CTMDPs, which appears difficult to achieve (unless resorting to IMCs right-away). Compositionality is not required for our purposes, where we target a transformation which converts a monolithic IMC into a monolithic CTMDP. The monolithic IMC itself is usually the result of a compositional construction, as exemplified in the workstation cluster example we shall consider later. The transformation from IMC to CTMDP is an *a posterior*-transformation only followed by a numerical analysis of the CTMDP.

As a consequence, we are able to view the entire IMC as a *closed* specification which is not interacting with the environment, but which exhibits the specified interactions (among parts of the specifications) to the environment. We therefore assume that interactive behaviour is *urgent* in the sense that it is assumed to consume no time, and is not blockable by the environment. This *closed-system* perspective is in contrast with the *open-system* perspective (which we employ during the construction of the IMC), where interaction may depend on interaction capabilities of the environment. This distinction has some implications on the formal setup we describe below, where the usual *maximal-progress* (interactions happen as soon as possible) hypothesis [6] is replaced by the above urgency hypothesis.

2.2 Interactive Markov chains

In the following we give the definitions of the interactive Markov chain model. We basically follow the notation in [6] where more details can be found.

Definition 1 (IMC). *An interactive Markov chain, IMC for short, is a tuple* $(S, Act, \longrightarrow, \dashrightarrow, s_0)$ *where* S *is a non-empty set of states,* Act *is a set of actions,* $\longrightarrow \subseteq S \times Act \times S$ *is a set of interactive transitions,* $\dashrightarrow \subseteq S \times \mathbb{R}^+ \times S$ *is a set of Markov transitions, and* $s_0 \in S$ *is the initial state.*

We use infix notation $s \xrightarrow{a} s'$ to denote $(s, a, s') \in \longrightarrow$ and $s \overset{\lambda}{\dashrightarrow} s'$ for $(s, \lambda, s') \in \dashrightarrow$.

Behaviour. An IMC can be viewed as a usual labelled transition system that additionally supports stochastic behaviour. The usual behaviour of labelled transition systems is provided by the interactive transitions leading from state s to s' via some action $a \in Act$. Stochastic behaviour is included via Markov transitions. A Markov transition leads from state s to s' with a particular rate $\lambda \in \mathbb{R}^+$. The delay of this transition is governed by a negative exponential distribution with rate λ, i. e., the probability of triggering this transition within t time units is given by $1 - e^{-\lambda \cdot t}$. When all transitions emanating from state s are Markov transitions, the next state is selected according to the *race condition* between these transitions. More precisely, assume that each of the states $u_1, u_2, \ldots, u_n$ is reachable from s via exactly one Markov transition with rate $\lambda_i, i = 1, 2, \ldots, n$, respectively. W.l.o.g.

we assume $u_1 = u_2 = \cdots = u_m = s', m \le n$. The probability to move from state s to state s' within t time units is then given as

$$\Pr(s, s', t) = \frac{\mathbf{Rate}(s, s')}{\mathbf{E}(s)} \left(1 - e^{-E(s)\cdot t}\right) \ ,$$

where $\mathbf{Rate}(s, s') = \sum_{i=1}^{m} \lambda_i$ and $\mathbf{E}(s) = \sum_{i=1}^{n} \lambda_i$ denote the cumulative rate from state s to s' and the total rate at which a transition emanating state s is taken, respectively. Suppose state s possesses both Markov and interactive outgoing transitions. In order to determine which transition is triggered it is in general not sufficient to take just Markov transitions into account, because an interactive transition might be taken beforehand. Interactive transitions may depend on triggers by the environment (which may occur with delay), in particular if the IMC under consideration is considered *open* for interaction. However, we have decided to stick to a *closed* interpretation. In this interpretation, interactive transitions are *urgent*, and thus happen instantaneously. They may thus be considered to have priority over Markov transitions.

As usual, we assume a distinguished internal action τ. We assume $\tau \in Act$, and let $Act_{\setminus\tau}$ denote $Act\setminus\{\tau\}$. The set of states of an interactive Markov chain can be partitioned into four disjoint subsets according to the outgoing transitions of each state. We distinguish:

- *Markov states* with at least one emanating Markov transition and no outgoing interactive transitions. Formally, S_M contains any state s satisfying $(\{s\} \times Act \times S) \cap \longrightarrow = \emptyset$ and $(\{s\} \times \mathbb{R}^+ \times S) \cap \dashrightarrow \neq \emptyset$.
- *Interactive states* with at least one emanating interactive transition and no outgoing Markov transitions. Formally, S_I contains any state s satisfying $(\{s\} \times Act \times S) \cap \longrightarrow \neq \emptyset$ and $(\{s\} \times \mathbb{R}^+ \times S) \cap \dashrightarrow = \emptyset$.
- *Hybrid states* with both Markov and interactive outgoing transitions, Formally, S_H contains any state s satisfying $(\{s\} \times Act \times S) \cap \longrightarrow \neq \emptyset$ and $(\{s\} \times \mathbb{R}^+ \times S) \cap \dashrightarrow \neq \emptyset$.
- *Sink states* without any outgoing transitions. Formally, S_{sink} contains any state s satisfying $(\{s\} \times Act \times S) \cap \longrightarrow = \emptyset$ and $(\{s\} \times \mathbb{R}^+ \times S) \cap \dashrightarrow = \emptyset$.

Hence, S can be written as $S = S_M \,\dot\cup\, S_I \,\dot\cup\, S_H \,\dot\cup\, S_{\text{sink}}$, where $\dot\cup$ denotes disjoint union. We do not consider orphan states which possess neither incoming nor outgoing transitions.

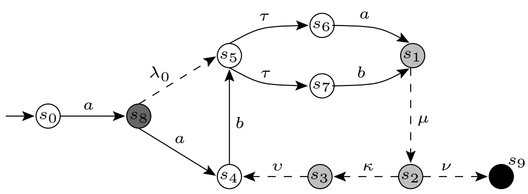

Fig. 2. Interactive Markov chain

Example 1. Figure 2 shows an example of an IMC. Solid arrows represent interactive transitions, dashed arrows correspond to Markov transitions. s_0 is the initial state. For this example, $S_M = \{s_1, s_2, s_3\}$, $S_I = \{s_4, s_5, s_6, s_7\}$ $S_H = \{s_8\}$, and $S_{\text{sink}} = \{s_9\}$. Different colours (white, light grey, dark grey, and black) are used to indicate interactive, Markov, hybrid and sink states. From hybrid state s_8, for example, either an a-transition can be taken to s_4 or s_5 can be reached via a Markov transition.

Equivalence notions. Strong and weak bisimulation equivalence are equivalence notions on interactive Markov chains. They will serve as correctness criteria for the model transformations we describe below. We state the definitions here and refer to [6] for more details. For an IMC $M = (S, Act, \longrightarrow, \dashrightarrow, s_0)$, we use the notation S_{IH} as an abbreviation for the set of states s which possess outgoing interactive transitions, i. e., $S_{IH} = S_I \dot{\cup} S_H$. For $C \subseteq S$ we denote by $\gamma(s, C)$, for $s \in S$, the total rate to move from state s to a state in C, i. e., $\gamma(s, C) = \sum\{|\lambda | s \overset{\lambda}{\dashrightarrow} s' \in C|\}$, where $\sum\{| \dots |\}$ denotes the sum of all elements in a multiset delimited by $\{| \dots |\}$.

Definition 2 (Strongly bisimilar IMCs). *An equivalence relation $\mathcal{E} \subseteq S \times S$ on an IMC $M = (S, Act, \longrightarrow, \dashrightarrow, s_0)$ is called a strong bisimulation iff $(s, u) \in \mathcal{E}$ implies*

1. *$s \xrightarrow{a} s'$ implies $u \xrightarrow{a} u'$ and $(s', u') \in \mathcal{E}$ for all $a \in Act$,*
2. *$s \notin S_{IH}$ implies $\gamma(s, C) = \gamma(u, C)$ for all equivalence classes C of $\mathcal{E}$.*

Two states s, u are strongly bisimilar, written $s \sim u$ if they are contained in some strong bisimulation $\mathcal{E}$. Two IMCs $M = (S, Act, \longrightarrow, \dashrightarrow, s_0)$, $M' = (S', Act, \longrightarrow', \dashrightarrow', s'_0)$ are strongly bisimilar, denoted $M \sim M'$, if $s_0 \sim s'_0$ on the disjoint union of S and S'.

Note, that the second condition requires the total rate to reach a certain equivalence class from state s to match the total rate from state u whenever state s possesses no outgoing interactive transitions. It differs from the definition in [6] where the second condition is imposed in the absence of internal τ-transitions from state s. The difference is justified by our *closed system* view, where we consider any interactive transition as urgent, regardless whether it is internal or not. This modification breaks the congruence property of $\sim$ for open (i. e., composable) systems.

We use $\overset{\tau}{\Longrightarrow}$ to denote the reflexive and transitive closure $\xrightarrow{\tau}^*$ of $\xrightarrow{\tau}$. For $a \in Act_{\setminus\tau}$, $\overset{a}{\Longrightarrow}$, denotes $\overset{\tau}{\Longrightarrow}\xrightarrow{a}\overset{\tau}{\Longrightarrow}$. We extend this to finite words $W = a_1 a_2 \dots a_n$ from $Act^+_{\setminus\tau}$ by letting $\overset{W}{\Longrightarrow}$ denote $\overset{a_1}{\Longrightarrow}\overset{a_2}{\Longrightarrow} \dots \overset{a_n}{\Longrightarrow}$. We recall the following definition of weak bisimilarity on interactive Markov chains from [6].

Definition 3 (Weakly bisimilar IMCs). *An equivalence relation $\mathcal{E} \subseteq S \times S$ on an IMC $M = (S, Act, \longrightarrow, \dashrightarrow, s_0)$ is called a weak bisimulation iff $(s, u) \in \mathcal{E}$ implies for all $a \in Act$ and all equivalence classes C of $\mathcal{E}$*

1. *$s \xrightarrow{a} s'$ implies $u \overset{a}{\Longrightarrow} u'$ for some u' and $(s', u') \in \mathcal{E}$,*
2. *$s \notin S_{IH}$ implies $\gamma(s, C) = \gamma(u', C)$ for some $u' \notin S_{IH}$ such that $u \overset{\tau}{\Longrightarrow} u'$.*

Two states s, u are weakly bisimilar, written $s \approx u$ if they are contained in some weak bisimulation $\mathcal{E}$. Two IMCs $M = (S, Act, \longrightarrow, \dashrightarrow, s_0)$, $M' = (S', Act, \longrightarrow', \dashrightarrow', s'_0)$ are weakly bisimilar, denoted $M \approx M'$, if $s_0 \approx s'_0$ on the disjoint union of S and S'.

Additionally we introduce a new equivalence on IMCs called *trace bisimilarity*. In Section 3.2 trace bisimilarity will serve as a correctness criterion for the transformation from IMCs to CTMDPs we shall define. In the following let $Words = Act^+_{\setminus\tau} \cup \{\tau\}$ and we use V, W, X to range over $Words$.

Definition 4 (Trace Bisimulation). *An equivalence relation $\mathcal{E}$ on an IMC $M = (S, Act, \longrightarrow, \dashrightarrow, s_0)$ is called trace bisimulation iff $(s, u) \in \mathcal{E}$ implies for all $W \in Words$*

1. *$s \overset{W}{\Longrightarrow} s'$ implies $u \overset{W}{\Longrightarrow} u'$,*

2. $s \stackrel{W}{\Longrightarrow} s' \notin S_{IH}$ *implies* $u \stackrel{W}{\Longrightarrow} u' \notin S_{IH}$ *and* $(s', u') \in \mathcal{E}$,
3. $s \notin S_{IH}$ *implies* $\gamma(s, C) = \gamma(u, C)$*, for all* $C \in S/\mathcal{E}$.

Two states s, u *are trace bisimilar, written* $s \sim_{tr} u$ *iff they are contained in some trace bisimulation* $\mathcal{E}$*. Two IMCs* $M = (S, Act, \longrightarrow, \dashrightarrow, s_0)$*,* $M' = (S', Act, \longrightarrow', \dashrightarrow', s'_0)$ *are trace bisimilar if* $s_0 \sim_{tr} s'_0$ *on the disjoint union of* S *and* S'.

Trace bisimulation ignores the branching structure among sequences of immediate transitions. Thus, for an ordinary LTS $M = (S, Act, \longrightarrow, \emptyset, s_0)$, trace bisimulation agrees with standard trace equivalence. For ordinary Markov chains $M = (S, Act, \emptyset, \dashrightarrow, s_0)$ on the other hand, trace bisimilarity agrees with strong bisimilarity, weak bisimilarity, and lumpability. In particular, it keeps the essentials of the branching structure between equivalence classes of $S_M / \sim_{tr}$.

Lemma 1. $\sim \subset \approx \subset \sim_{tr}$.

2.3 Continuous-time Markov Decision Processes

In the following we present the basic definitions of continuous-time Markov decision processes. We use essentially the same notation as the one presented in [14], except that the transitions of a CTMDP are defined by means of a transition relation $\mathbf{R}$, instead of a transition function.

Definition 5 (CTMDP). *A continuous-time Markov decision process (CTMDP) is a tuple* $(S, L, \mathbf{R}, s_0)$ *where* S *is a non-empty set of states,* L *is a set of transition labels also called actions,* $\mathbf{R} \subseteq S \times L \times (S \longrightarrow \mathbb{R}_{\geq 0})$ *is the set of transitions,* $s_0 \in S$ *is the initial state.*

Each CTMDP transition can be viewed as a labelled *directed hyperedge* [21]. A directed hyperedge is an ordered pair (X, Y) where X and Y are subsets of states indicating the tails and heads of the hyperedge. Note that the hyperedge is directed from tails to heads. For a given hyperedge $E = (X, Y)$, we use $t(E)$ to refer to its tails X and $h(E)$ to refer to its heads Y. Each transition $(s, A, R) \in \mathbf{R}$ in a CTMDP corresponds to a hyperedge E with $t(E) = \{s\}$ and $h(E) = \{s' \mid R(s') > 0)\}$; thus it can be seen as mono-tailed hyperedge where the tail of a transition is labelled by action A and heads are labelled by the positive elements of R.

Behaviour. The behaviour of a CTMDP is as follows. Suppose $(s, A, R) \in \mathbf{R}$ for some $R : S \longrightarrow \mathbb{R}_{\geq 0}$. Then $R(s') > 0$ indicates the existence of a transition emanating from s and leading to s' labelled with A. If there are distinct outgoing transitions of s, one of them is selected nondeterministically. This nondeterminism is solved by a *scheduler*, sometimes also called *adversary* or *policy*. Given that a transition labelled A has been selected, the probability of triggering that A-transition from s to s' within t time units equals $1 - e^{-R(s') \cdot t}$. If $R(s') > 0$ for more than one s' the transition (s, A, R) can be viewed as a hyperedge labelled with A, connecting s with target states s', and possibly different rates for each of the s'. In this case there is a race between the relevant transitions. For a more detailed description see [14].

Example 2. Figure 3 shows an example CTMDP with five states and initial state t_0. It contains six transitions, five of which are trivial hyperedges, since they connect singleton heads and tails. State t_2, for example, allows for two nondeterministic alternatives – labelled C and D – both leading with rate μ to t_1. The only nontrivial hyperedge connects t_1 and $\{t_3, t_4\}$. The tail is labelled by T. The branch leading to t_3 (respectively t_4) is labelled by κ (ν).

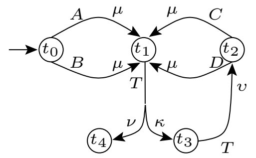

Fig. 3. Continuous-time Markov decision process

Equivalence. We introduce the notion of strong bisimilarity on CTMDPs as follows. If $(s, A, R) \in \mathbf{R}$, we denote by $R(C)$ for $C \subseteq S$ the total rate to move from state s to some state in C via action A, that is, $R(C) = \sum_{s' \in C} R(s')$.

Definition 6 (Strongly bisimilar CTMDPs). *An equivalence relation $\mathcal{E} \subseteq S \times S$ on a given CTMDP $M = (S, L, \mathbf{R}, s_0)$ is a strong bisimulation iff $(s, u) \in \mathcal{E}$ implies for each $A \in L$ that whenever $(s, A, R) \in \mathbf{R}$ there is some R' such that $(u, A, R') \in \mathbf{R}$ and $R(C) = R'(C)$ for each equivalence class C of $\mathcal{E}$.*

Two states s, u are strongly bisimilar, written $s \sim u$ if they are contained in some strong bisimulation $\mathcal{E}$. Two CTMDPs $M = (S, L, \mathbf{R}, s_0)$, $M' = (S', L, \mathbf{R}', s_0')$ are strongly bisimilar, denoted $M \sim M'$, if $s_0 \sim s_0'$ on the disjoint union of S and S'.

This is a straightforward adaption of the notion of strong bisimulation on (non-)probabilistic transition systems, Markov chains and discrete-time Markov decision processes [22]. Besides being straightforward, it is unclear to us whether this notion has appeared in the scientific literature so far. Remark, that $\sim$ is overloaded – it will be clear from context if it is used as an equivalence symbol on IMCs or CTMDPs.

3 Transformation Procedure

In this section we provide the essential transformation steps that are used to transform an IMC into a CTMDP.

It is important to note that we do not aim at resolving nondeterminism but in collapsing nondeterministic choices. The goal of our transformation is to get a CTMDP that corresponds in its timed behaviour to the input IMC. Resolving nondeterminism in the resulting CTMDP will be done according to different types of schedulers but is not in the scope of this paper since we are not facing at analysing CTMDPs. In order to keep the timed behaviour one of the transformation steps will flatten the branching structure at interactive states. As will become clear this is necessary in order to end up in the desired CTMDP formalism.

Additionally, we provide a justification for each of these steps by showing that each step leading from IMC to CTMDP preserves strong bisimulation. Figure 4 gives an overview. Each solid arrow is labelled with the preservation property of the corresponding transformation step. For example, the transformation from *IMC* to *alternating IMC* preserves strong bisimilarity (in both directions).

Model restrictions. We impose some restrictions on the model of interactive Markov chains prior to its transformation. The IMCs considered do not contain cycles of interactive transitions. In other words, the transition relation $\longrightarrow^*$ is acyclic. This is a technical restriction which can be relinquished by resorting to the algebraic theory of IMC [6]. Together with the *urgency* assumption, the above acyclicity assumption avoids Zeno-behaviour, where

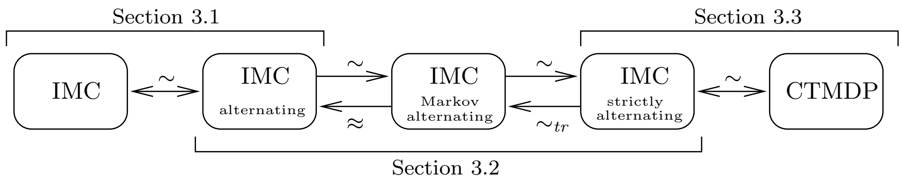

Fig. 4. Transformation steps and preservation properties

infinite computation may happen in finite or zero time. For convenience, we assume that $s_0 \in S_I$. Additionally, we restrict ourselves to IMCs where the set S_{sink} is assumed to only be reachable from Markov states:

$$(S_M \times (Act \cup \mathbb{R}^+) \times S_{\text{sink}}) \cap (\longrightarrow \cup \dashrightarrow) = (S \times (Act \cup \mathbb{R}^+) \times S_{\text{sink}}) \cap (\longrightarrow \cup \dashrightarrow) .$$

The degree to which these restrictions can be circumvented will be discussed in Section 5.

3.1 Identification of States

In this first step the interactive Markov chain is turned into an *alternating* interactive Markov chain. That is, all states are classified as either Markov or interactive states. This means that hybrid and sink states will be re-classified.

Definition 7 (Alternating IMC). *An interactive Markov chain with state space S is called alternating iff none of the states $s \in S$ possesses both Markov and interactive outgoing transitions, i. e., for all $s \in S$, $(\{s\} \times \mathbb{R}^+ \times S) \cap \dashrightarrow \neq \emptyset$ implies $(\{s\} \times Act \times S) \cap \longrightarrow = \emptyset$.*

Hybrid states. Owed to our *urgency* assumption, which is motivated by our closed system view, hybrid states are simple to handle. From a stochastic perspective, the probability that an exponential delay finishes instantaneously is zero. On the other hand, interactive transitions are *urgent*, and thus happen instantaneously. This observation justifies to assume that a hybrid state performs interactive transition only and is not allowed to let time pass. As a consequence, all hybrid states will be interpreted as interactive states, i. e., for a state $s \in S_H$ all Markov transitions emanating from state s are cut off and s is added to S_I.

Sink states. It remains to handle sink states. Due to the above restriction, the set S_{sink} is assumed to only be reachable from Markov states. Therefore we can simply consider any sink state as an interactive state. To achieve this, we alter the definition of interactive states slightly. For an IMC $M = (S, Act, \longrightarrow, \dashrightarrow, s_0)$, we say state $s \in S$ is an interactive state iff $(\{s\} \times \mathbb{R}^+ \times S) \cap \dashrightarrow = \emptyset$, and we let – from now on – S_I denote the set of all such states. This ensures $S_{\text{sink}} \subseteq S_I$.

Transformation. More formally, this transformation step can be summarised as follows.

Definition 8 (Transformation $\mathbf{a}(\cdot)$). *Let $M = (S, Act, \longrightarrow, \dashrightarrow, s_0)$ be an IMC. We define $\boldsymbol{a}(M) = (S, Act, \longrightarrow, \dashrightarrow_{\boldsymbol{a}(M)}, s_0)$ as the corresponding alternating IMC with $\dashrightarrow_{\boldsymbol{a}(M)} = (S_M \times \mathbb{R}^+ \times S) \cap \dashrightarrow$.*

Note, that due to the restriction of $\dashrightarrow_{\mathbf{a}(\cdot)}$ to Markov states all Markov transitions emanating from a state $s \in S_H$ in M are cut in $\mathbf{a}(M)$. The state space S of the obtained IMC equals the one of the input IMC; but is now the disjoint union of S_M and S_I', where the latter set comprises original sink states (S_{sink} in M) and original hybrid states (S_H in M), besides original interactive states.

Lemma 2. *Let M be an interactive Markov chain. Applying Definition 8 to M results in an alternating IMC $\mathbf{a}(M)$.*

Example 3. Consider the IMC depicted in Figure 2. The transformation to an alternating IMC yields the one shown in Figure 5. State s_8 has now become an interactive one by removing transition $s_8 \overset{\lambda_0}{\dashrightarrow} s_5$. State s_9 is added to the set of interactive states and is no more characterised as an (interactive) sink state.

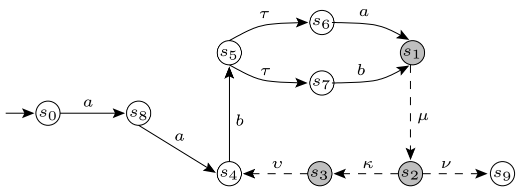

Fig. 5. Alternating IMC

As stated in the following proposition, transforming an interactive Markov chain M into its alternating counterpart $\mathbf{a}(M)$ preserves strong bisimulation.

Proposition 1 (Alternating IMC) *Let M and N be two IMCs and denote by $\mathbf{a}(M)$ and $\mathbf{a}(N)$ their alternating counterparts. Then: $\mathbf{a}(M) \sim \mathbf{a}(N)$ iff $M \sim N$.*

3.2 Strictly Alternating

The second step is divided into two parts. In summary, the interactive Markov chain will be turned into a *strictly alternating* interactive Markov chain, i. e., in an alternating IMC in which each Markov state has only interactive states as predecessors and successors and, vice versa, each interactive state has only Markov states as predecessors and successors.

Definition 9 (Strictly alternating IMC). *Let $M = (S, Act, \longrightarrow, \dashrightarrow, s_0)$ be an alternating IMC, i. e., $S = S_I \dot{\cup} S_M$. Iff $(S_M \times \mathbb{R}^+ \times S_M) \cap \dashrightarrow = \emptyset$ and $(S_I \times Act \times S_I) \cap \longrightarrow = \emptyset$, we call M strictly alternating. An IMC satisfying the first (respectively second) condition is called Markov (interactive) alternating IMC.*

To structure our discussion we mention that the strict alternation property is violated if sequences of either Markov states or interactive states are contained in the model. First, we give a description of how to eliminate sequences of Markov states and afterwards we describe our approach to resolve sequences of interactive states.

Markov sequences. The resolution of sequences of Markov states is rather straightforward. As interactive transitions consume no time, interactive transitions can be inserted in-between two consecutive Markov transitions. This implies to insert new interactive states, but the newly inserted states are only capable of a *poor man's choice*, that is, the non-determinism boils down to just one possible decision. The following definition makes the transformation process precise.

Definition 10 (Transformation $\mathbf{mA}(\cdot)$). *Let $M = (S, Act, \longrightarrow, \dashrightarrow, s_0)$ be an alternating IMC. The transformation $\boldsymbol{mA}(M) := (S', Act, \longrightarrow_{\boldsymbol{mA}(M)}, \dashrightarrow_{\boldsymbol{mA}(M)}, s_0)$, is given by*

- $S' = S \,\dot\cup\, \{(s, s') \in S_M \times S_M \mid \exists \lambda \in \mathbb{R}^+ : s \overset{\lambda}{\dashrightarrow} s'\}$,
- $\longrightarrow_{\boldsymbol{mA}(M)} = \longrightarrow \;\dot\cup\; \{((s, s'), \tau, s') \in S' \times \{\tau\} \times S_M \mid \exists \lambda \in \mathbb{R}^+ : s \overset{\lambda}{\dashrightarrow} s'\}$,
- $\dashrightarrow_{\boldsymbol{mA}(M)} = \dashrightarrow \cap\, (S_M \times \mathbb{R}^+ \times S_I) \;\dot\cup\; \{(s, \lambda, (s, s')) \in S_M \times \mathbb{R}^+ \times S' \mid s \overset{\lambda}{\dashrightarrow} s'\}$.

As a result of applying transformation $\mathbf{mA}(\cdot)$ to an alternating IMC M we obtain the following lemma.

Lemma 3. *Let M be an alternating IMC. Applying Definition 10 to M results in a Markov alternating IMC $\boldsymbol{mA}(M)$.*

Example 4. When we apply $\mathbf{mA}(\cdot)$ to the IMC depicted in Figure 5 we obtain the IMC shown in Figure 6. This IMC is Markov alternating. s_1' and s_2' are the newly inserted interactive states.

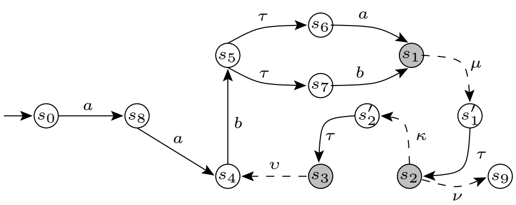

Fig. 6. Markov alternating IMC

The following proposition states that $\mathbf{mA}(\cdot)$ preserves strong bisimulation in one direction.

Proposition 2 (Markov alternating IMC) *Suppose the alternating IMCs M and N are given. It holds*

$$M \sim N \textit{ implies } \boldsymbol{mA}(M) \sim \boldsymbol{mA}(N), \quad \textit{and} \quad \boldsymbol{mA}(M) \approx \boldsymbol{mA}(N) \textit{ implies } M \approx N.$$

Note, that strong bisimulation can not be established for the second implication. For the two weak bisimilar IMCs $s_1 \overset{\lambda}{\dashrightarrow} s_2 \overset{\tau}{\longrightarrow} s_3 \overset{\mu}{\dashrightarrow} s_4$ and $u_1 \overset{\lambda}{\dashrightarrow} u_2 \overset{\mu}{\dashrightarrow} u_3$ the transformation yields strong bisimilar Markov alternating IMCs.

Interactive sequences. In case of sequences of interactive states we need to proceed in a different way. Due to our closed system view, sequences of interactive transitions can happen in zero time without requiring any interaction. These sequences must be of finite length, since the interactive transition relation is assumed to be acyclic (and this acyclicity is trivially preserved by the steps discussed so far).

We define a transformation based on the transitive closure of the interactive transition relation. For each interactive state s that has at least one Markov state as direct predecessor and at least one interactive state as direct successor, we determine the Markov states which terminate all these sequences. This means that the transitive closure on the interactive transition relation is calculated in a fashion that returns all of these Markov states. These states are used to define a strictly alternating IMC where interactive transitions are labelled by words of $Act^{+}_{\setminus\tau}$ and always end in a Markov state. Interactive states possessing only interactive states as direct predecessors (except the initial state) are removed from the resulting state space. The following definition makes the *transitive closure transformation* step precise.

Definition 11 (Transformation $\mathbf{iA}(\cdot)$). *Let $M = (S, Act, \longrightarrow, \dashrightarrow, s_0)$ be a (Markov) alternating IMC. The transformation $\boldsymbol{iA}(M) := (S', Words, \longrightarrow_{\boldsymbol{iA}(\cdot)}, \dashrightarrow, s_0)$, is given by*

- $S' = S_M \mathbin{\dot\cup} S'_I$ *where* $S'_I = \{s \in S_I \mid \exists t \in S_M : t \overset{\lambda}{\dashrightarrow} s$, *for some* $\lambda \in \mathbb{R}^+\} \cup \{s_0\}$,
- $\longrightarrow_{\boldsymbol{iA}(\cdot)} := \{(s, W, t) \in S'_I \times Words \times S_M \mid s \overset{W}{\Longrightarrow} t\}$.

Recall that *Words* is defined as $Act^{+}_{\setminus\tau} \cup \{\tau\}$.

Lemma 4. *Let M be a Markov alternating IMC then $\boldsymbol{iA}(M)$ is the corresponding strictly alternating IMC.*

Example 5. The IMC depicted in Figure 7 is obtained from the one shown in Figure 6 when transformation $\mathbf{iA}(\cdot)$ is applied. States s_5, s_6, s_7 and s_8 do not possess Markov predecessors and hence they are not present in the strictly alternating IMC. State s_4, for example, now possesses two outgoing transitions leading with ba and bb to s_1, a Markov state.

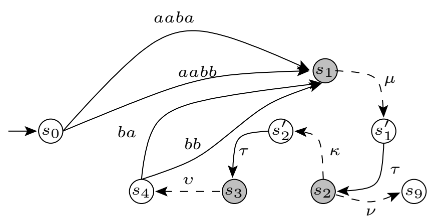

Fig. 7. Strictly alternating IMC labelled with words

The transformation realised by $\mathbf{iA}(\cdot)$ preserves strong bisimulation. This result is stated in the following proposition.

Proposition 3 (Strictly alternating IMC) *Suppose M and N are (Markov) alternating IMCs. It holds*

$$M \sim N \text{ implies } \boldsymbol{iA}(M) \sim \boldsymbol{iA}(N), \quad \text{and} \quad \boldsymbol{iA}(M) \sim_{tr} \boldsymbol{iA}(N) \text{ implies } M \sim_{tr} N.$$

3.3 IMC as CTMDP

As a result of the previous transformation steps we obtain a strictly alternating interactive Markov chain. A strictly alternating IMC M can be regarded as a continuous-time Markov decision process as follows. Suppose that in M are two interactive states s and v with $s \xrightarrow{A} u$ and $u \overset{\lambda}{\dashrightarrow} v$ for some Markov state u, i. e., s is a direct predecessor of u and v a direct successor of u. If action A is selected in s then v is reachable via u within t time units with probability $1 - e^{-\lambda \cdot t}$. In other words, the selection of an action corresponds to the selection of a particular timed behaviour of the IMC. As motivated in Section 2, this is exactly the interpretation of the behaviour of a CTMDP. The following definition formalises this interpretation; basically it shows how to construct a CTMDP out of a strictly alternating IMC. We will see that this construction establishes an isomorphism between strictly alternating IMCs and CTMDPs.

Definition 12 (Interpretation $\Phi(\cdot)$). *Let $M = (S, Words, \longrightarrow, \dashrightarrow, s_0)$ be a strictly alternating interactive Markov chain. The underlying CTMDP $\Phi(M) = (\widetilde{S}, Words, \mathbf{R}, s_0)$, is given by $\widetilde{S} = S_I$, and $\mathbf{R} = \{(s, A, R) | R(s') = \sum_{i=1}^{n} \lambda_i$ iff $\exists u \in S_M, \lambda_i \in \mathbb{R}_{\geq 0}$ such that $s \xrightarrow{A} u \wedge u \overset{\lambda_i}{\dashrightarrow} s', i = 1, 2 \ldots, n\}$.*

Example 6. Consider the IMC depicted in Figure 7. We can interpret this strictly alternating IMC as a CTMDP. Assume the following abbreviations. $A = aab, B = aabb, C = ba, D = bb, T = \tau$. With these abbreviations in mind, the interpretation as CTMDP yields the CTMDP shown in Figure 3 if states are renamed accordingly, i. e., s_0 becomes t_0, s_1' becomes t_1, s_4 becomes t_2, s_2' becomes t_3, s_9 becomes t_4.

The following proposition states that strictly alternating interactive Markov chains and their underlying continuous-time Markov decision processes coincide under strong bisimulation.

Proposition 4 (IMC/CTMDP) *Let M and N be two strictly alternating IMCs with their underlying CTMDPs $\Phi(M)$ and $\Phi(N)$ respectively. Then: $M \sim N$ iff $\Phi(M) \sim \Phi(N)$.*

In summary we have established the following correspondence between IMC and CTMDP. The corollary states that if two IMCs M and N are strongly bisimilar then the CTMDPs obtained from their strictly alternating counterparts are strongly bisimilar.

In the following we refer to a CTMDP $\Phi(\mathbf{iA}(\mathbf{mA}(\mathbf{a}(N))))$ obtained from an IMC N as the *underlying CTMDP of* N, denoted $\widetilde{N}$ for short. Note, that in $\widetilde{N}$, transition labels are finite words over the transition labels of N, owed to the transitive closure in Section 3.2. In summary we obtain the following result.

Corollary 1. *Suppose M, N are interactive Markov chains then*

$$M \sim N \quad \textit{implies} \quad \widetilde{M} \sim \widetilde{N}, \qquad \textit{and} \qquad \widetilde{M} \sim \widetilde{N} \quad \textit{implies} \quad M \sim_{tr} N.$$

3.4 Time and space requirements

It is interesting to study the time and space requirements of the transformation algorithm, as well as the size of the underlying CTMDP of an IMC with respect to the number of states and transitions. From the described transformation steps we may conclude the following theorem.

Theorem 1 (Space and time complexity). *Let* $M = (S, Act, \longrightarrow, \dashrightarrow, s_0)$ *be an interactive Markov chain and* $\widetilde{M}$ *its underlying CTMDP* $(\widetilde{S}, Words, \mathbf{R}, s_0)$. *The number of states of* $\widetilde{M}$ *is at most* $|S|$. *The number of hyperedges of* $\widetilde{M}$ *is at most* $| \longrightarrow |$, *where for each hyperedge* E *contained in* $\widetilde{M}$ $|h(E)|$ *is at most* $| \dashrightarrow |$. *The entire transformation requires* $\mathcal{O}(|S|^3)$ *time and* $\mathcal{O}(|S|^2)$ *space.*

The worst case time complexity is dominated by the transitive closure operation, where we assume a standard (cubic) implementation, and do not consider specialised algorithms, such as [23].

4 Experimental evaluation

In this section we discuss some details of an on-the-fly implementation of the algorithm, and report on the application of this implementation to the workstation cluster example mentioned in the introduction.

4.1 An on-the-fly implementation

To implement the transformation from IMC to CTMDP we explore the state space of the IMC $(S, Act, \longrightarrow, \dashrightarrow, s_0)$ in a breadth-first-search manner, starting from the initial state s_0. For each new state s we encounter we scan its emanating transitions to decide its type, which can be one of the following.

- *Interactive state:* If state s is reached via a Markov state, it will remain in the resulting CTMDP. We implement the required transitive closure of interactive transitions emanating s in a depth-first-search manner. We explore sequences of interactive transitions $s \xrightarrow{a} s'$ until we find a Markov state s'' while collecting the word W labelling this sequence. This implicitly covers the case of interactive states which are not reached via a Markov state. The label W will label the tail of the hyperedge connecting s and the successors of s''. We continue the DFS exploration of the interactive subgraph at s until all words W and Markov states s'' are found.
- *Markov state:* Each Markov state s encountered gives rise to a hyperedge, where the heads of this hyperedge are given by the Markov transitions $s \overset{\lambda}{\dashrightarrow} s'$. The tail and the source state have been determined before. The targets of these heads are the successors s' of s, except if s' is a Markov state itself. In this case, an auxiliary interactive state s'' is assumed between the two states in the IMC under exploration, as described in Section 3.2.
- *Hybrid state:* Hybrid states are directly transformed into and handled as interactive states.
- *Sink state:* Sink states can (by assumption) be treated as interactive states. But since they do no possess any outgoing transitions they are just target states of hyperedges, there is no further computation required.

This on-the-fly procedure resembles algorithms used for computing well-definedness/well-specifiedness [10, 11]. It differs since it does not incorporate probabilities and is tailored to cases where nondeterminism is present in the model, i. e., where well-definedness/well-specifiedness fails.

4.2 Fault tolerant workstation cluster

To experiment with a prototypical implementation of the on-the-fly transformation procedure, we apply the transformation to a *fault tolerant workstation cluster* described in [8], which is modelled by the confused GSPN depicted in Figure 1. The cluster itself consists of two sub-clusters each with N workstations (LeftWorkStationUp, RightWorkStationUp) that are connected with a central switch (LeftSwitchUp, RightSwitchUp). The switches provide the interface to the backbone (BackboneUp). Each of the components can fail. A single repair unit (RepairUnitAvailable) is available for repairing defect components. We refer to [8] for a more detailed description.

We developed a model of the workstation cluster in the algebra of IMC, and compositionally generated a lumped IMC using the LOTOS-toolkit CADP [24]. The modelling is straightforward, given the GSPN in Figure 1. The construction of the lumped IMC is more interesting, but is not in the scope of the paper. We refer to [25, 26] for the strategy used, which alternates construction and weak bisimulation aggregation steps, and is implemented in CADP. We constructed lumped IMCs for varying numbers N of workstations in the left and right cluster.

Applying the transformation procedure proposed above to these IMC yields the results depicted in Table 1. The first columns display various statistics of the input IMCs, namely parameter N, the number of interactive states, Markov states, hybrid states, interactive transitions and Markov transitions, respectively. The next four columns comprise the respective information for the strictly alternating IMC generated as a result of the on-the-fly IMC transformations. Note that the strictly alternating IMC does not contain any hybrid states anymore. Information about the corresponding CTMDP is contained in the five rightmost columns. The first of these columns shows the number of states – which is by construction identical to the number of interactive states in the strictly alternating IMC. The second and third column show the overall number of hyperedge tails and heads, while the last two columns give the maximal number of hyperedge tails and heads emanating a state.

The maximal number of hyperedge tails per state corresponds to the *degree of nondeterminism* in the system, i. e., the maximal number of nondeterministic choices in a state. The maximal degree of nondeterminism is constant 3. This is owed to situations where the repair unit must choose between failed workstations, switches, and backbones. Note that in our model (as in [8]) all workstations have identical rates, and also both switches have the same rate. The resulting model symmetry is a priori exploited during the IMC construction, thus only three essentially different choices remain. If multiple different rates were used, or if the model symmetry were not exploited, the degree of nondeterminism would increase accordingly.

As can be seen from Table 1, the observed model sizes are far below the worst-case expectations given in Theorem 1. The computation time required for each of the transformations is below one second on a standard Linux workstation.

5 Discussion and Conclusion

This paper has investigated the relation between interactive Markov chains and continuous-time Markov decision processes. We briefly discuss some features of the transformation and some directions for future work.

Preservation of properties. In a nutshell we have developed a transformation from IMCs to CTMDPs which transforms strongly bisimilar IMCs to strongly bisimilar CTMDPs. Yet

	Interactive Markov chain					Strictly alternating IMC				CTMDP				
													max. per state	
N	S_I	S_M	S_H	$\longrightarrow$	$\dashrightarrow$	S_I	S_M	$\longrightarrow$	$\dashrightarrow$	$\widetilde{S}$	tails	heads	tails	heads
1	13	34	4	33	118	47	34	61	88	47	61	159	3	9
2	23	60	6	59	222	85	60	113	166	85	113	315	3	9
4	37	112	16	111	430	161	112	217	322	161	217	627	3	9
8	65	216	36	215	846	313	216	425	634	313	425	1251	3	9
16	121	424	76	423	1678	617	424	841	1258	617	841	2499	3	9
32	233	840	156	839	3342	1225	840	1673	2506	1225	1673	4995	3	9

Table 1. Transformation results

some structure is lost during the transformation because the branching structure among sequences of immediate moves has no counterpart in the CTMDP world. Therefore, we have that if two strong bisimilar CTMDPs are images of two IMCs, then the latter are ensured to be trace bisimilar, not strong or weak bisimilar.

One may argue, that the notion of trace bisimilarity, albeit being simple and intuitive, is a novel relation and as such does not give sufficient insight into the properties preserved by the transformation. We here give a brief practical account of the correspondence of the two models. Due to space constraints we stay on an intuitive level, and use the notion of a *scheduler* $\mathcal{S}$ as a means to resolve nondeterminism in IMCs or CTMDPs in some appropriate manner. We further take an observational perspective, where an observer performs repetitive experiments looking at the finite traces (observed sequences of visible actions) exhibited by the system (which can be a scheduled IMC $\mathcal{S}(M)$ or scheduled CTMDP $\mathcal{S}'(\widetilde{M})$), possibly making statistics about the likelihood and timing of the actions witnessed. It is not difficult to show that, for each scheduler $\mathcal{S}$ of an IMC M (resp. CTMDP $\widetilde{M}$), there is a scheduler $\mathcal{S}'$ of the CTMDP $\widetilde{M}$ (resp. IMC M) ensuring the following three properties.

- Any trace $W \in Act^+_{\setminus\tau}$ can be observed on $\mathcal{S}(M)$ iff it can be observed on $\mathcal{S}'(\widetilde{M})$.
- The probability of observing any trace $W \in Act^+_{\setminus\tau}$ agrees on $\mathcal{S}(M)$ and on $\mathcal{S}'(\widetilde{M})$.
- For any trace $W \in Act^+_{\setminus\tau}$, the delay elapsing between an arbitrary pair (a, b) of visible actions appearing in $W = TaUbV$, for arbitrary traces T, U, V, follows the same phase-type distribution on $\mathcal{S}(M)$ and on $\mathcal{S}'(\widetilde{M})$.

This holds for deterministic and randomized, for history-dependent and history-independent, and for time-abstract and time-dependent schedulers.

Relinquishing initial restrictions. We briefly discuss in how far our initial restrictions made on the general model of interactive Markov chains can be relinquished.

(1) We restricted to models without cycles of interactive transitions. Such a cycle corresponds to an infinite computation taking zero-time. From an algebraic perspective, this can be equated with a time-lock [6]. One may imagine a time-locking extension of CTMDPs, but we see no practical applications of this concept and therefore abstain from its inclusion in the model transformation. (2) The initial state is so far assumed to be interactive. This restriction is straightforward to relinquish (Markov states are treated by prefixing with a poor-man's choice interactive state). (3) We imposed the restriction that sink states can only be reached via Markov predecessors. Sink states with interactive predecessors are not easy to handle, because the corresponding CTMDP behaviour may need to terminate right after taking the last decision, without letting any further (exponentially distributed) time

pass. This would mean to catch the CTMDP behaviour 'in the middle' of a transition which is not straightforward.

Further work. This work is motivated by the practical problem how to analyse IMCs with nondeterminism, 'non-well specified' SAN and 'confused' GSPNs. The transformation we established enables us to apply CTMDP analysis algorithms, as described, e. g., in [14,19] to these models, thus easing the use of the above modelling formalisms, where so far models often were considered to resist any kind of analysis. Since we currently do not support discrete probability distributions over states, its use is restricted to SANs with singleton cases, and GSPNs without weight assignments. To relinquish this restriction while assuring a semantically sound transformation is ongoing work.

Another direction of further work is to devise a temporal logic formalism with semantically consistent interpretation over both IMCs and CTMDPs. It appears natural to strive for an appropriate combination of an action-based logic (such as ACTL [27]), and a stochastic logic (such as CSL [28]).

We do not plan to extend our transformation to open IMCs. While this would be a theoretically nicer approach to analysing IMCs, it implicitly requires to turn CTMDPs itself into a compositional formalism. But since in CTMDPs actions and timing (in the form of rates) are 'glued' together, it appears to us that a proper way of handling compositionality for CTMDPs will lead to a formalism very close to the well-developed compositional theory of IMC. We believe it is more adequate to perform compositional model construction on the IMC level (as is done in the workstation cluster example) followed by two monolithic steps: transformation to a CTMDP, and subsequent analysis of the CTMDP. Another promising alternative is to lift the CTMDP analysis algorithms to IMCs.

Acknowledgements. The authors are grateful to Joost-Pieter Katoen (RWTH Aachen/University of Twente) for detailed comments on an earlier version of this paper. This work is partially supported by the NWO-DFG bilateral project Validation of Stochastic Systems (VOSS), the NWO Vernieuwingsimpuls award 016.023.010 (VoPaD), and the DFG special research center SFB/TR 14 (AVACS).

References

1. Ajmone Marsan, M., Balbo, G., Conte, G., Donatelli, S., Franceschinis, G.: Modelling with Generalized Stochastic Petri Nets. (1995)
2. Sanders, W.H., Meyer, J.F.: Stochastic Activity Networks: Formal Definitions and Concepts. In: European Educational Forum: School on Formal Methods and Performance Analysis, LNCS 2090. (2001) 315–343
3. Hillston, J.: A Compositional Approach to Performance Modelling. Cambridge University Press (1996)
4. Bravetti, M., Bernardo, M.: Compositional Asymmetric Cooperations for Process Algebras with Probabilities, Priorities, and Time. In: Proceedings MTCS. Volume 39(3). (2000)
5. Hermanns, H., Herzog, U., Mertsiotakis, V.: Stochastic Process Algebras - Between LOTOS and Markov Chains. Computer Networks & ISDN Systems **30**(9-10) (1998) 901–924
6. Hermanns, H.: Interactive Markov Chains and the Quest for Quantified Quality. LNCS 2428. Springer (2002)
7. Haverkort, B.R.: Approximate performability and dependability analysis using generalized Petri nets. Performance Evaluation **18** (1993) 61–78
8. Haverkort, B.R., Hermanns, H., Katoen, J.P.: On the use of model checking techniques for dependability evaluation. In: Proc. 19th Symp. on Reliable Distributed Systems, 16-18 October 2000, Nürnberg, Germany. (2000) 228–237
9. Chiola, G., Donatelli, S., Franceschinis, G.: GSPNs versus SPNs: What is the actual role of immediate transitions? In: Proceedings PNPM. (1991) 20–31

10. Ciardo, G., Zijal, R.: Well-defined stochastic Petri nets. In: Proceedings of the 4th International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems, IEEE Computer Society (1996) 274
11. Deavours, D.D., Sanders, W.H.: An efficient well-specified check. In: Proceedings PNPM. (1999) 124–133
12. Bertsekas, D.P.: Dynamic Programming and Optimal Control. Volume II. Athena Scientific (1995)
13. Feinberg, E.A.: Continuous Time Discounted Jump Markov Decision Processes: A Discrete-Event Approach. Mathematics of Operations Research **29** (2004) 492–524
14. Puterman, M.L.: Markov decision processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons (1994)
15. Sennot, L.: Stochastic Dynamic Programming and the Control of Queueing Systems. John Wiley & Sons (1999)
16. Bruno, J., Downey, P., Frederickson, G.N.: Sequencing Tasks with Exponential Service Times to Minimize the Expected Flow Time or Makespan. J. ACM **28**(1) (1981) 100–113
17. Abdeddaïm, Y., Asarin, E., Maler, O.: On Optimal Scheduling under Uncertainty. TACAS (2003) 240–253
18. Qiu, Q., Pedram., M.: Dynamic power managment based on continuous-time Markov decision processes. In: Proceedings DAC. (1999) 555 – 561
19. Baier, C., Haverkort, B.R., Hermanns, H., Katoen, J.P.: Efficient Computation of Time-Bounded Reachability Probabilities in Uniform Continuous-Time Markov Decision Processes. In: Proceedings TACAS. (2004)
20. de Alfaro, L.: Formal verification of probabilistic systems. PhD thesis, Stanford University (1997)
21. Gallo, G., Longo, G., Pallottino, S., Nguyen, S.: Directed hypergraphs and applications. Discrete Appl. Math. **42**(2-3) (1993) 177–201
22. Givan, R., Dean, T., Greig, M.: Equivalence notions and model minimization in Markov decision processes. Artif. Intell. **147**(1-2) (2003) 163–223
23. Coppersmith, D., Winograd, S.: Matrix multiplication via arithmetic progressions. In: Proceedings of Conference on Theory of computing. (1987) 1–6
24. Garavel, H., Lang, F., Mateescu, R.: An Overview of CADP 2001. In: EASST Newsletter. Volume 4. (August 2002)
25. Hermanns, H., Katoen, J.P.: Automated compositional Markov chain generation for a plain-old telephone system. Sci. Comput. Program. **36**(1) (2000) 97–127
26. Garavel, H., Hermanns, H.: On Combining Functional Verification and Performance Evaluation Using CADP. In: Proceedings FME. (2002) 410–429
27. R. de Nicola, F. Vandraager: Three logics for branching bisimulation. ACM **42**(2) (March 1995) 458–487
28. Baier, C., Haverkort, B.R., Hermanns, H., Katoen, J.P.: Model-Checking Algorithms for Continuous-Time Markov Chains. IEEE Trans. Software Eng. **29**(6) (2003) 524–541