Analysing reward measures of LARES performability models by discontinuous Markov chains

Alexander Gouberman*, Martin Riedl and Markus Siegle

Institut für Technische Informatik
Universität der Bundeswehr München
Werner-Heisenberg-Weg 39, 85579 Neubiberg, Germany
Email: alexander.gouberman@unibw.de
Email: martin.riedl@unibw.de
Email: markus.siegle@unibw.de
*Corresponding author

Abstract: This paper presents a new method for specifying and analysing Markovian performability models. An extension of the LARES modelling language is considered which offers both delayed and immediate transitions, as well as rate and impulse rewards on whose basis different types of reward measures can be defined. The paper describes the evaluation path, starting from the modular and hierarchical LARES description and leading via a flat labelled transition system to the underlying stochastic model. The latter is a continuous-time Markov chain with fast transitions which, by taking the limit of the fast transition rates, can be interpreted as a CTMC with stochastic discontinuities. Finally, by continuisation of impulse rewards and an aggregation process, a standard Markov reward model is obtained.

Keywords: performability, Markov reward model, impulse reward, immediate transition, CTMC with stochastic discontinuities, LARES

Reference to this paper should be made as follows: xxxx (xxxx) 'xxxx', *xxxx*, Vol. x, No. x, pp.xxx–xxx.

Biographical notes:
Alexander Gouberman studied mathematics with a focus on differential geometry at Universität Augsburg. Since 2008 he is a research assistant for the chair "Design of Computer and Communication Systems" at the department of computer science of Universität der Bundeswehr München. His research interests lie in the field of evaluation and optimization of dependability and performance measures for stochastic systems.

Martin Riedl studied computer science with compiler construction and syntax analysis as area of specialization at Friedrich-Alexander Universität Erlangen-Nürnberg. As of 2007 he is a research assistant for the chair "Design of Computer and Communication Systems" at the department of computer science of Universität der Bundeswehr München. Apart from the definition of the LARES language and its underlying semantics, he is mainly involved in the continuous development of the associated transformation and analysis framework.

Markus Siegle is professor for "Design of Computer and Communication Systems" at the department of computer science of Universität der Bundeswehr München. He studied computer science and electrical engineering at Stuttgart

University, was a Fulbright student at North Carolina State University and obtained the doctoral and habilitation degrees from Erlangen University. His research is on the modelling, analysis and optimisation of probabilistic and non-deterministic systems, with special focus on quantitative aspects such as performance and dependability. He has worked intensively on data structures and tools for analysing very large Markov models, as well as model checking of non-functional requirements.

1 Introduction

Dependability metrics such as availability and survivability are crucial for modern ICT systems, especially in the area of critical infrastructures. System architects seek to achieve these features by providing for fault-tolerance and automatic dynamic reconfiguration. In order to assess dependability-related properties, Markov reward models (MRM) can be employed, from which a wide range of measures of interest can be computed. The LARES modelling language ("LAnguage for REconfigurable Systems", first described in [Gouberman et al., 2009]) with its associated toolset offers a comfortable and powerful environment for the generation and analysis of MRMs.

LARES provides means to define elementary behaviour in the form of labelled automata, and self-contained modules comprising behaviour or submodule instantiations. Abstract definitions of behaviours and modules can be instantiated which enables the modeller to specify clearly structured (i.e. modular and hierarchic) models with clear visibility and scoping semantics. Complex interactions between modules may be specified, by triggering a specific reaction if a composed state satisfies a certain condition. Hereby, the reaction may involve non-trivial synchronisation.

The transformation semantics of LARES to a stochastic process algebra (SPA), which serves as a low-level evaluation formalism, has been sketched in [Riedl and Siegle, 2012]. The present paper is an extended version of the workshop paper [Gouberman et al., 2013], albeit with different focus. Its main contributions are as follows: We consider a very general class of LARES.re MRMs (where "re" stands for reward extension) with Markovian and immediate transitions, with rate and impulse rewards and different analysis types for the reward measures. For this class of models, we define the semantics by a sequence of transformations, first to a continuous-time Markov reward model with fast transitions ($\mathrm{CTMRM}_{\mathrm{fast}}$), and from there further to a continuous-time Markov reward model with stochastic discontinuities ($\mathrm{CTMRM}_{\mathrm{disc}}$). In particular, it is shown that this methodology provides a sound semantics for models containing immediate transitions associated with impulse rewards.

Related Work: Among earlier approaches to the modelling of rewards in a Markovian context we mention Stochastic Reward Nets [Muppala et al., 1994], Stochastic Activity Networks [Qureshi et al., 1996] and the models accepted by the model checker MRMC [Katoen et al., 2011]. In comparison with these, LARES models are at a higher level of abstraction, which offers much more structure and flexibility, thus easing the modelling process. LARES and its reward extension are intended to be used as engineering formalisms on a similar level as, for instance, AltaRica [Point, 2000] or the SLIM language [Bozzano et al., 2009b]. Similar to LARES, AltaRica and SLIM offer means for hierarchical modelling. While AltaRica is a mature language with many extensions focusing mainly on functional verification, SLIM aims at detecting flaws in early design-stages of hardware/software systems in an automated manner. For SLIM models, it can be checked whether they meet quantitative requirements specified with the help of the logics PCTL or CSL. For that purpose, SLIM models are transformed to their underlying Markov chain which is then checked with the model checker MRMC [Bozzano et al., 2009a]. Compared to these approaches, the distinguishing feature of LARES is an easy specification of combinatorial aspects spanning across multiple hierarchy levels. In addition to that, the extension LARES.re offers a very expressive specification of rate rewards and impulse rewards which is fully integrated with LARES' concepts of modularity and hierarchy. These concepts enable a very clear and flexible modelling of rewards whose value may depend on complex conditions. For GSPNs without rewards, [Ammar et al., 1987] proposed a method of analysis based on stochastically discontinuous Markov processes, which is related to the approach described in this paper.

Organisation of the paper: The paper is organised as follows: Following this introduction, Sec. 2 recapitulates the theoretical background, providing the basic definitions and properties of continuous-time Markov reward models (CTMRMs), Markov chains with stochastic discontinuities and Markov chains with fast transitions. Sec. 3 introduces the reward extension LARES.re of the language by means of a running example. In Sec. 4 we sketch the LTS semantics on the basis of which the reachable state space is constructed. We omit the formal definition of the semantics, since that can be found in the previous paper [Gouberman et al., 2013]. Sec. 5 discusses the transformation (involving multiple steps) of the LTS to an analysable MRM in which the immediate transitions (which may carry impulse rewards) have been eliminated. Finally, Sec. 6 concludes the paper.

2 Theoretical Background

2.1 Continuous Time Markov Reward Models

Notation: For $n \in \mathbb{N}$ let $\mathcal{S} := \{1, \dots, n\}$ be a finite state space. We consider probability distributions over $\mathcal{S}$ as row vectors in $\mathbb{R}^{1\times n}$ and define $\mathcal{D}_n \subseteq \mathbb{R}^{1\times n}$ as the set of all probability distributions over $\mathcal{S}$. Let δ_{ij} denote the Kronecker-δ and for $s \in \mathcal{S}$ let $\mathbf{1}_s \in \mathbb{R}^n$ be the column vector with $(\mathbf{1}_s)_{s'} := \delta_{ss'}$ such that $\mathbf{1}_s^T \in \mathcal{D}_n$ and define $\mathbf{1} := \sum_s \mathbf{1}_s$. For a matrix $A \in \mathbb{R}^{n\times n}$ define $\Delta(A) \in \mathbb{R}^n$ as its diagonal and for a vector $v \in \mathbb{R}^n$ define $\mathrm{Diag}(v) \in \mathbb{R}^{n\times n}$ as the diagonal matrix with diagonal v.

A *continuous-time Markov chain* (CTMC) over $\mathcal{S}$ is an $\mathcal{S}$-valued stochastic process $(X_t)_{t\geq 0}$ and its probabilistic behavior is uniquely characterized by a transition matrix function $P : [0, \infty) \to \mathbb{R}^{n\times n}$ that fulfills for all $s, t \geq 0$ the properties

$$P(t) \geq 0, \quad P(t)\mathbf{1} = \mathbf{1} \quad \text{and} \quad P(s+t) = P(s)P(t) \text{ with } P(0) = I, \tag{1}$$

i.e. $P(t)$ is a stochastic matrix and fulfills the Chapman-Kolmogorov equation. From (1) it follows that $P(t)$ is continuous for all $t > 0$. In literature one often demands as a further regularity condition the continuity at $t = 0$, i.e. $\lim_{t\to 0} P(t) = P(0)$. In this case the CTMC is called *regular* and $P(t) = e^{Qt}$ where $Q = \lim_{t\to 0} \frac{1}{t}(P(t) - I) \in \mathbb{R}^{n\times n}$. The matrix Q has zero row sums and nonnegative off-diagonal entries that can be regarded as rates of exponentially distributed times for transition. In a more general setting, if the regularity condition is not assumed in addition to (1), then $P(t) = \Pi e^{Qt}$ where $\Pi = \lim_{t\to 0} P(t)$ and $Q = \lim_{t\to 0} \frac{1}{t}(P(t) - \Pi)$ [Coderch et al., 1983]. The matrices Π and Q fulfill

$$\Pi \geq 0, \quad \Pi\mathbf{1} = \mathbf{1}, \quad \Pi^2 = \Pi, \tag{2}$$

$$\Pi Q = Q\Pi = Q, \quad Q\mathbf{1} = 0 \quad \text{and} \quad Q + c\Pi \geq 0 \text{ for some } c \geq 0. \tag{3}$$

Moreover, every such tuple (Π, Q) that fulfills (2) and (3) uniquely characterizes a CTMC by setting $P(t) := \Pi e^{Qt}$. The matrix Π is referred to as the *ergodic projection at zero* and Q as the *generator matrix*. If $\Pi \neq I$ then $P(t)$ has a discontinuity at $t = 0$ and X_t is a stochastically discontinuous process with sample paths that almost surely have an infinite number of jumps within a finite time interval. On the other hand, if $\Pi = I$ then X_t is a regular CTMC that is stochastically continuous and its number of jumps within finite time is almost surely finite. We additionally consider a CTMC with an initial distribution $\sigma \in \mathcal{D}_n$ over states such that a CTMC is given by the triple (Π, Q, σ) and a regular CTMC by the tuple (Q, σ). The more general discontinuous CTMCs are considered in the forthcoming Sec. 2.2. For the remainder of this subsection, we stick to regular CTMCs.

A regular *continuous-time Markov reward model* (CTMRM) over $\mathcal{S}$ is a quadruple (Q, σ, r, i) that consists of a regular CTMC (Q, σ), a rate reward vector $r \in \mathbb{R}^n$ over states and an impulse reward matrix $i \in \mathbb{R}^{n\times n}$ over transitions with $i_{ss} = 0$ for all s. Let a path ω of X_t be represented by $\omega := s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} s_2 \xrightarrow{t_2} s_3 \ldots$ where s_i is a visited state and $t_i > 0$ the sojourn time in s_i. Along ω the total accumulated reward up to time $T > 0$ is given by $\sum_{k\leq N_T} \left(r_{s_k} t_k + i_{s_k s_{k+1}}\right)$ where N_T is the number of performed transitions up to time T. The *finite-horizon total value function* $V_T \in \mathbb{R}^n$ (represented as a vector) assigns for each state s the expected value $(V_T)_s$ of accumulated rewards over all paths starting in s. In order to compute V_T we can transform the CTMRM (S, Q, r, i) into a CTMRM $(S, Q, \overline{r})$ without impulse rewards [Gouberman and Siegle, 2014]. Hereby the impulse reward $i_{ss'}$ are weighted with the corresponding rates $Q_{ss'}$ and merged with r into

$$\overline{r}_s := r_s + \sum_{s'\neq s} i_{ss'} Q_{ss'}. \tag{4}$$

We refer to $\overline{r}$ as the *continuized* rate reward. Since $i_{ss} = 0$ we can also write $\overline{r}$ in vector notation as $\overline{r} = r + \Delta(Qi^T)$, where $\Delta(M) \in \mathbb{R}^n$ is the diagonal of the matrix M. We will stick to the CTMRM structure with impulse rewards, since it is more natural to distinguish time-based rewards from time-independent rewards, especially in the context of modular model specification. If $R(t) := P(t)\overline{r} \in \mathbb{R}^n$ (with $P(t) = e^{Qt}$) denotes the expected rate reward at point in time $t \geq 0$ then the finite-horizon total value vector for the CTMRM $(Q, \overline{r})$ is given by

$$V_T = \int_0^T R(t)\, dt \tag{5}$$

and the finite-horizon total value for the initial distribution σ by $\sigma V_T \in \mathbb{R}$. In the following we briefly outline the most important infinite-horizon reward measures that are based on V_T [Guo and Hernandez-Lerma, 2009].

(i) The *total value function* $V_\infty \in \mathbb{R}^n$ is the expectation over all paths of all rewards accumulated along these paths over infinite time. If this accumulation process absolutely converges, then $V_\infty = \lim_{T\to\infty} V_T$ and it is the unique solution to

$$QV_\infty = -\overline{r} \tag{6}$$

with the property that $(V_\infty)_s = 0$ for all recurrent states s.

(ii) The *α-discounted value function* $V^\alpha := \int_0^\infty e^{-\alpha t} R(t)\, dt \in \mathbb{R}^n$ with $\alpha > 0$ is the accumulation of rewards along all paths where rewards gained in the future are continuously discounted with rate α. It holds that the accumulation process converges for all $\alpha > 0$ and V^α is the unique solution to

$$(Q - \alpha I)V^\alpha = -\overline{r}. \tag{7}$$

(iii) The *average value function* $g := \lim_{T\to\infty} \frac{1}{T} V_T$ measures the accumulated reward over all paths averaged over the infinite time horizon. Let $P^* := \lim_{t\to\infty} P(t)$ be the limiting distribution matrix, i.e. the row corresponding to the s-th state is the limiting distribution of the CTMRM when starting in s. Then

$$g = P^*\overline{r}. \tag{8}$$

2.2 Stochastically Discontinuous CTMRMs

In this section we briefly outline the probabilistic semantics of a discontinuous CTMC (Π, Q) with state process X_t. For general Π that fulfills (2) and (3), the generator Q can have negative off-diagonal entries in contrast to the case of regular CTMCs. [Coderch et al., 1983] showed that Π partitions the state space into ergodic classes at zero $\mathcal{E}_0^k$, $k = 1, \ldots, K$ and transient states at zero $\mathcal{T}_0$ such that $\mathcal{S} = \bigcup_{k=1}^K \mathcal{E}_0^k \cup \mathcal{T}_0$. It further holds that Π has the canonical product decomposition $\Pi = RL$, where $L \in \mathbb{R}^{K\times n}$ contains the ergodic probability distributions (of the K ergodic classes at zero) and $R \in \mathbb{R}^{n\times K}$ contains the trapping probabilities into the ergodic classes. The value $\Pi_{ss'}$ can be regarded as the probability to find the system in state s' immediately after entering state s since $\lim_{h\to 0} P(X_{t+h} = s' \mid X_t = s) = \Pi_{ss'}$. The ergodic class $\mathcal{E}_0^k$ can be seen as a macro state containing all of its states $s \in \mathcal{E}_0^k$.

Consider the case that $\mathcal{E}_0^k$ has at least two states. If at some point in time $t \geq 0$ the process X_t enters (an arbitrary state of) the class $\mathcal{E}_0^k$ and the state of X_t is measured at time t, then this state is not deterministically predictable. Instead, one can only find the system in a certain state $s' \in \mathcal{E}_0^k$ with probability $\Pi_{ss'} > 0$, where $s \in \mathcal{E}_0^k$ is arbitrary. This means that if state s is entered at time t, then it is immediately redistributed among states $s' \in \mathcal{E}_0^k$ according to the probabilities $\Pi_{ss'}$. This behavior is reflected in the sample paths $\omega : [0, \infty) \to \mathcal{S}$ of the process that have almost surely an infinite number of jumps between the states in $\mathcal{E}_0^k$. If there is a timed transition from some state $s \in \mathcal{E}_0^k$ to a state $s' \in \mathcal{S} \setminus \mathcal{E}_0^k$ given by the generator Q, then the ergodic class at zero $\mathcal{E}_0^k$ can be almost surely left within finite time. In case $\mathcal{E}_0^k$ contains only one state s then s is called regular and corresponds to a state of a regular Markov chain.

A discontinuous CTMC (Π, Q, σ) can be reduced to a regular CTMC $(\widehat{Q}, \widehat{\sigma})$ with generator $\widehat{Q} := LQR \in \mathbb{R}^{K\times K}$ and initial distribution $\widehat{\sigma} := L\sigma$. Hereby the ergodic classes at zero $\mathcal{E}_0^k$ are aggregated to regular states and all transient states at zero are eliminated. The transition function of the reduced CTMC satisfies $\widehat{P}(t) = LP(t)R$ and can be disaggregated back by $P(t) = R\widehat{P}(t)L$. Thus, from a probabilistic point of view a discontinuous CTMC with state process X_t and its aggregated version are the same modulo reduction. However, when we are concerned with modelling the uncertainty of states, the formalism of discontinuous CTMCs is more appropriate then that of regular CTMCs. Furthermore, we will see in Section 2.3 another advantage of discontinuous CTMCs as a limiting process of Markov chains with fast transitions.

In order to deal with rewards, the discontinuous CTMC (Π, Q, σ) can be enhanced to a discontinuous CTMRM (Π, Q, σ, r) with a rate reward vector $r \in \mathbb{R}^n$ [Markovski et al., 2009]. The reward function is given by $R(t) := P(t)r$ and its s-th component $R_s(t)$ denotes the expected reward that is gained at time t if s is the initial state. Note that $P(t) = P(t)\Pi$ and thus the rate reward r can be modified to Πr without having an influence on the reward function $R(t)$. This means that in an ergodic class $\mathcal{E}_0^k$ the gained rate reward $R_s(t)$ is constant for all $s \in \mathcal{E}_0^k$ and it is the weighted sum of the rate rewards $r_{s'}$ with weights $\Pi_{ss'}$. In analogy to (5), the finite-horizon total value is given by $V_T := \int_0^T R(t)\,dt$. The reduction of (Π, Q, σ) to $(\widehat{Q}, \widehat{\sigma})$ induces the according reduction of the rate reward r (or Πr) to $\widehat{r} := Lr$. Furthermore, the reduction of (Π, Q, σ, r) to $(\widehat{Q}, \widehat{\sigma}, \widehat{r})$ and the evaluation of the reward function commute since $\widehat{R}(t) = \widehat{P}(t)\widehat{r} = (LP(t)R)(Lr) = L(P(t)r) = LR(t)$.

2.3 Continuous Time Markov Reward Models with Fast Transitions

A CTMC with fast transitions is a triple (S, F, σ) where $S \in \mathbb{R}^{n\times n}$ and $F \in \mathbb{R}^{n\times n}$ are rate matrices of slow (or rare) and fast transitions and $\sigma \in \mathcal{D}_n$ an initial distribution. In this subsection we consider the case that $\Delta(S) = 0$ and $\Delta(F) = 0$, i.e. there are no self-loops on states. We will relax this restriction in Section 2.4. Let $Q^S := S - \text{Diag}(S\mathbf{1})$ and $Q^F := F - \text{Diag}(F\mathbf{1})$ denote the corresponding regular generator matrices. Consider for each $\tau \geq 0$ the regular generator matrix $Q_\tau := Q^S + \tau Q^F$ with transition function $P_\tau(t) := e^{Q_\tau t}$. The probabilistic semantics of (S, F, σ) is defined as the limiting behavior of $P_\tau(t)$ as $\tau \to \infty$. It can be shown that $P_\tau(t)$ converges pointwise in $t \in [0, \infty)$ and locally uniformly on $(0, \infty)$ to the transition function $P(t) := \Pi e^{Qt}$ of a discontinuous CTMC (Π, Q, σ) where the ergodic projection at zero $\Pi := \lim_{\tau\to\infty} e^{Q^F \tau}$ is the limiting matrix of the fast transitions F (i.e. the ergodic projection at infinity of F) and the generator matrix is given by $Q := \Pi S \Pi$ [Coderch et al., 1983, Theorem 4.3]. We denote the transient resp. ergodic states of the CTMC F as F-*transient* resp. F-*ergodic* states. Thus, the decomposition of the state space by Π into transient states and ergodic classes at zero is precisely the decomposition by F into F-transient states and F-ergodic classes.

[Markovski et al., 2009] extended the notion of a CTMC with fast transitions by a rate reward $r \in \mathbb{R}^n$ and defined its semantics by transformation to a discontinuous CTMRM (Π, Q, σ, r). The semantics can be alternatively given in form of a limit of $R_\tau(t) := P_\tau(t)r$ for $\tau \to \infty$ and coincides with the semantics of the discontinuous CTMRM. For the purpose of this paper we consider a CTMRM with fast transitions $\mathcal{C} := (S, F, \sigma, r, i)$ that also includes an impulse reward matrix $i \in \mathbb{R}^{n\times n}$ over transitions. The semantics of $\mathcal{C}$ regarding the reward accumulation process is given by the finite-horizon total value V_T which we are going to define in the following.

As an example, consider Figure 1. From initial state 1 a slow transition with rate λ_{12} can be performed and an impulse reward i_{12} is gained. State 2 is F-transient and there can be

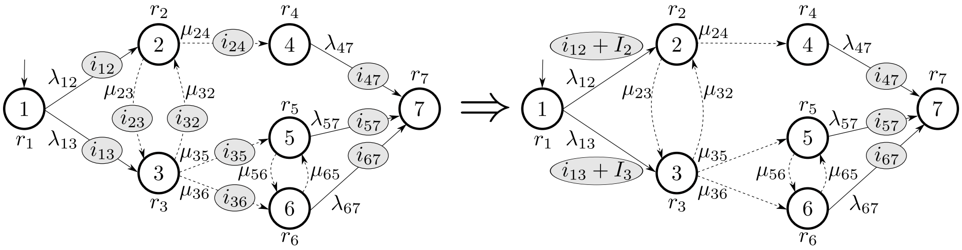

Figure 1 Moving impulse rewards from fast transitions to impulse rewards on slow transitions by computing the infinite-horizon total value function, e.g. $I_2 = \left(V_\infty^F\right)_2$.

cycles with state 3 along which the impulse rewards i_{23} and i_{32} are accumulated until one of the F-ergodic classes $\mathcal{E}_0^1 := \{4\}$ or $\mathcal{E}_0^2 := \{5, 6\}$ is reached. The expected accumulated impulse rewards from state 2 until getting trapped in some F-ergodic class is given by the infinite-horizon total value $I_2 := \mathbf{1}_2^T V_\infty^F \in \mathbb{R}$, where $V_\infty^F = \int_0^\infty e^{Q^F u} \Delta(F i^T)\, du$. Note that we have implicitly assumed that V_∞^F is finite, and this is the case if there are no impulse rewards on transitions between recurrent states in the same recurrent class (as for states 5 and 6 in $\mathcal{E}_0^2$). Otherwise, if V_T^F diverges with $T \to \infty$ we define the model $\mathcal{C}$ as invalid. The total sojourn time along F-transient states until absorption in an F-ergodic class is 0 with probability 1. Therefore, we can regard I_2 as an impulse reward that is gained when some F-ergodic class $\mathcal{E}_0^i$ is reached from state 2, i.e. I_2 is gained at the point in time of arrival in $\mathcal{E}_0^i$. Since the semantics of a fast-transition CTMC is given by a discontinuous CTMC, the point in time of the transition from state 1 to state 2 coincides with the point in time of departure in state 1 and the point in time of arrival in some F-ergodic class. Thus, we can alternatively accumulate the impulse reward I_2 by an arrival to state 2 from state 1 and therefore integrate I_2 into the impulse reward for transition to state 2, i.e. modify the impulse reward i_{12} to $i_{12} + I_2$. This first intermediate transformation results in a CTMRM $\mathcal{C}_{\text{int}}$ with no impulse rewards for fast transitions and thus only for slow transitions. In a second step we transform $\mathcal{C}_{\text{int}}$ to a CTMRM $\mathcal{C}_{\text{final}} = (S, F, \sigma, \overline{r})$ without impulse rewards by continuizing all impulse rewards into a rate reward, e.g. for state 1 it holds that $\overline{r}_1 = r_1 + \lambda_{12}(i_{12} + I_2) + \lambda_{13}(i_{13} + I_3)$. Since $\mathcal{C}_{\text{final}}$ has no impulse rewards it can be analysed by its underlying discontinuous CTMRM $\mathcal{C}_{\text{disc}} := (\Pi, Q, \sigma, \overline{r})$. However, this transformation is not yet complete, since there can be an initial state in the support of σ that is F-transient. In such a state the accumulated impulse rewards up to absorption in some F-ergodic class can be included into the model $\mathcal{C}_{\text{final}}$ by defining additionally an initial reward R_0 that is gained when such an F-ergodic class is reached. As above, it holds that $R_0 = V_\infty^F$. Collecting all together, the finite-horizon total value function V_T is given for $T > 0$ by

$$V_T := V_\infty^F + \int_0^T \Pi e^{Qt} \overline{r}\, dt, \quad \text{where} \quad \overline{r} := r + \Delta(S i^T) + S V_\infty^F. \tag{9}$$

The s-th component in V_T is the total expected accumulated reward in the interval $[0, T]$ from initial state s. Thus, the assigned value for the CTMRM model $\mathcal{C}$ is given by the real-valued function $T \mapsto \sigma V_T \in \mathbb{R}$ for $T \geq 0$. Note that $V_0 := 0$ and $\lim_{T \to 0} V_T = V_\infty^F$ such that V_T is continuous at $T = 0$ if and only if $V_\infty^F = 0$. This is the case if and only if there are no impulse rewards on fast transitions (since we assumed that the reward accumulation

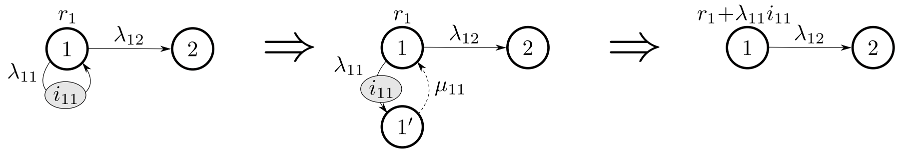

Figure 2 Processing of impulse rewards on self-loops by means of an intermediate fast-transition CTMRM with an auxiliary F-transient state and its subsequent elimination by reduction.

over infinite time converges absolutely). We refer to $\overline{r}$ as the *continuized rate reward* and it involves the following three contributions for the reward accumulation process: the rate reward r for states, the impulse rewards i for slow transitions continuized into $\Delta(Si^T)$ and the accumulated impulse rewards for fast transitions continuized into the rate reward SV_∞^F. Note that instead of $\overline{r}$ we can alternatively consider the rate reward $\Pi\overline{r}$, since the accumulation of rate rewards over F-transient states is 0 and over F-ergodic classes they can be averaged with respect to the stationary distributions in the classes. This has the implication that there are also other possible equivalent definitions for a continuized rate reward $\overline{r}'$ as long as $\Pi\overline{r} = \Pi\overline{r}'$. As an example, since $(V_\infty^F)_s = 0$ for every F-ergodic state s it follows that $\Pi\,\mathrm{Diag}(S\mathbf{1})\,V_\infty^F = 0$ and thus $\Pi SV_\infty^F = \Pi Q^S V_\infty^F$ such that one can consider the continuized rate reward $\overline{r}' := r + \Delta(Si^T) + Q^S V_\infty^F$ which leads to the same total value function V_T.

2.4 Adding self-loops

In this section we consider a CTMRM model $\mathcal{C} = (S, F, \sigma, r, i)$ where S and F are allowed to have self-loops, i.e. these rate matrices can have positive rates on their diagonal. Although self-loops do not have any effect on the probabilistic behavior of the system, they are necessary if we are concerned with the accumulation of impulse rewards for self-loops. Note that the generator matrices $Q^S := S - \mathrm{Diag}(S\mathbf{1})$ and $Q^F := F - \mathrm{Diag}(F\mathbf{1})$ do not involve the rates for such self-loop transitions. As in the previous section, we can transform $\mathcal{C}$ to a discontinuous CTMRM $\mathcal{C}_{\text{disc}} = (\Pi, Q, \sigma, \overline{r})$ with

$$\overline{r} := r + \Delta(Si^T) + SV_\infty^F, \quad \text{where} \quad V_\infty^F := \int_0^\infty e^{Q^F u}\Delta(Fi^T)\,du.$$

In the following, we are going to justify this relation. For each self-loop we can extend the state space with an auxiliary F-transient state as shown in Figure 2. The transitions are rebuilt in such a way that both the probabilistic behavior of the model and the reward accumulation process are not influenced. If we order the states in the sequence $(1, 1', 2)$ then

$$\Pi = \begin{pmatrix} 1\,0\,0 \\ 1\,0\,0 \\ 0\,0\,1 \end{pmatrix}, \quad Q = \begin{pmatrix} -\lambda_{12}\; 0\; \lambda_{12} \\ -\lambda_{12}\; 0\; \lambda_{12} \\ 0 \quad 0 \quad 0 \end{pmatrix}, \quad L = \begin{pmatrix} 1\,0\,0 \\ 0\,0\,1 \end{pmatrix} \quad \text{and} \quad R = \begin{pmatrix} 1\,0 \\ 1\,0 \\ 0\,1 \end{pmatrix}.$$

A further elimination of the auxiliary intermediate state by reduction yields

$$\widehat{Q} := LQR = \begin{pmatrix} -\lambda_{12}\; \lambda_{12} \\ 0 \quad 0 \end{pmatrix} = Q^S, \quad \widehat{\overline{r}} := L\overline{r} = \begin{pmatrix} r_1 + \lambda_{11} i_{11} \\ 0 \end{pmatrix} = r + \Delta(Si^T).$$

Thus, one can eliminate self-loops and thereby continuize the impulse reward on such a transition by weighting it with the corresponding rate. If there are self-loops on F-transient states then impulse rewards on those are accordingly accumulated by considering the fast transitions separately as slow transitions for the regular CTMRM $(F, \sigma, 0, i)$. With same arguments as above, the continuized rate reward is given by $\Delta(Fi^T)$ and therefore the finite-horizon total value for the original fast-transition CTMRM by $V_\infty^F = \int_0^\infty e^{Q^F u} \Delta(Fi^T)\, du$.

3 LARES.re (Reward Extension)

A reward structure can be defined in LARES inside a Module definition. A rate reward can be modelled by a StateReward statement that consists of a state reward expression, which is an arithmetic expression and extends arithmetic atoms (number value or reference to some parameter) by state reward atoms. A state reward atom can be either an indicator $[\mathtt{A}]$ over some condition `A` or a reference to another state reward statement (within its visibility scope). The indicator $[\mathtt{A}]$ is evaluated to either 1 or 0 depending on whether the condition `A` is satisfied or not. An impulse reward is specified by a TransitionReward statement that differs from a StateReward statement only in the indicator $[\mathtt{A} \to \mathtt{B}]$ that needs both a precondition `A` and a postcondition `B`. Since indicators can be evaluated to 0 we do not allow to divide reward expressions by other reward expressions. A RewardMeasure statement assigns to a state reward or a transition reward (or a tuple consisting of both) one of the reward analysis types total, discounted or average.

Before showing the semantics of LARES.re in Section 5 (by a transformation into a CTMRM), we first present a running example that among others comprises deadlocks. The transformation maps these deadlocks to stochastic discontinuities and thus resolves them in a probabilistical way.

3.1 Running Example

Figure 3 provides the specification of the running example used throughout this paper. It shows a system `main` (line 42) which consists of two instances `C1` and `C2` representing the components of the system. The container component `C2` contains two subcomponents `SC1` and `SC2` (line 26). Each of these basic components `C1`, `SC1` and `SC2` inherits from a behavior `B` (line 13) which provides the necessary states and transitions for the failure and repair behavior (lines 1..11). From an `active` state, a component may fail with rate `0.1`, and it has the capability to heal itself with rate `0.01`. Alternatively, the component can move immediately via the ⟨`rep`⟩ guard label to the `inRep` state, which denotes an ongoing repair process (with the rate `mu` set to the value `2.0`) that runs in parallel to the self-healing process. The repair process of each component can be stopped by the ⟨`stop`⟩ guard label. In contrast to the basic components, which fail if their behavior is not active (defined by the condition `failed`, line 16), the container component `C2` is a series network that fails if one of its subcomponents fails (line 31). The whole system is a parallel network of `C1` and `C2` (line 47). By the Initial statements, the initial state for each instantiated behavior is set to `active`.

The first guards statement (lines 49-52) is responsible for the repair process, which will be triggered (only) in case the condition `C1.failed` is satisfied. The reactive part `C1.`⟨`rep`⟩ triggers the forward label ⟨`rep`⟩ in the `Component` module, which in turn triggers the ⟨`rep`⟩ guard label in the behavior `B`. In case `C2` has also already failed, the repair is processed in

```
1  Behavior B(mu) {
     State active, failed, inRep
     Transitions from active
       if ⟨true⟩ → failed, delay exponential 0.1
     Transitions from failed
6      if ⟨true⟩ → active, delay exponential 0.01
       if ⟨rep⟩ → inRep, weight 1.0
     Transitions from inRep
       if ⟨true⟩ → active, delay exponential mu + 0.01
       if ⟨stop⟩ → failed, weight 1.0
11 }

   Module Component : B(mu=2.0) {
     Initial init = B.active

16   Condition failed = !B.active
     Condition inRep = B.inRep

     forward ⟨rep⟩ to B.⟨rep⟩
     forward ⟨stop⟩ to B.⟨stop⟩
21
     StateReward energy = 2.5∗[B.active]
     TransitionReward rst = 5.0∗[B.inRep→B.active]
   }

26 Module Container {
     Instance SC1 of Component
     Instance SC2 of Component
     Initial init = SC1.init, SC2.init

31   Condition failed = SC1.failed | SC2.failed
     Condition inRep = SC1.inRep | SC2.inRep

     forward ⟨rep⟩ to maxsync{SC1.⟨rep⟩,SC2.⟨rep⟩}
     forward ⟨stop⟩ to maxsync{SC1.⟨stop⟩,SC2.⟨stop⟩}
36
     StateReward cEnergy = SC1.energy + SC2.energy
     StateReward energy = 0.9∗cEnergy + 0.5
     TransitionReward rst = 6.0∗[failed → !failed]
   }
41
   System main {
     Instance C1 of Component
     Instance C2 of Container
     Initial init = C1.init, C2.init
46
     Condition failed = C1.failed & C2.failed

     C1.failed guards {
       C1.⟨rep⟩
51     C2.⟨rep⟩ if C2.failed
     }

     C1.inRep & C2.inRep guards
       sync{C1.⟨stop⟩, C2.⟨stop⟩}
56
     StateReward energy = C1.energy + C2.energy
     RewardMeasure M1 = energy discounted 0.01

     TransitionReward rst = C1.rst + C2.rst +
61                          10.0∗[failed → !failed]
     RewardMeasure M2 = (energy, rst) average
   }
```

Figure 3 Running example: LARES.re model

the form of a choice: C2.⟨rep⟩ triggers in parallel to C1.⟨rep⟩ the forward label ⟨rep⟩ in the Container module which is responsible for the repair of both of its subcomponents SC1 and SC2 by way of a maximal synchronisation. All subcomponents which are failed are thus put to repair synchronously, i.e. all of them move to the inRep state immediately in one transition. If both C1 and C2 are in repair then the total repair process will be overloaded and can break down for both components synchronously. This behavior is modelled by the second guards statement (lines 54-55). Hereby, all components that are in repair are triggered to be failed immediately such that the first guards statement can be activated again. This can lead to a repair of both C1 and C2 which induces in turn an overload of the repair process. As a consequence, this repair-and-stop process induces a deadlock which however can be resolved probabilistically by the semantics of a discontinuous CTMRM.

We assume that every component consumes energy and that we want to measure the global energy consumption of the system. A basic component continuously consumes 2.5 units of energy per time unit if it is active (line 22). If the repair process of a component is successful, an impulse reward of 5.0 energy units for the restart is consumed (line 23). The term cEnergy inside the Container module describes the energy consumed by its subcomponents over time, and the total energy decreases the energy consumption of its components by 10% and adds its own consumption of `0.5`. Furthermore, when restarting the `Container` there is only energy cost for the container itself but not for its subcomponents (line 39). In the `System` definition, we finally define the total `energy` of its components (line 57).

The reward measure `M1` specifies all the ingredients that are necessary in order to compute the discounted consumption of the continuous energy for the whole system with a discount rate of `0.01`. This discount rate can be interpreted as a rate for an exponential distribution that describes the random time length (horizon) in which the rewards are accumulated. Therefore, the expected horizon length for reward accumulation is 100 time units. Note that since the system can get repaired, the total energy consumption over an infinite horizon without discounting would lead to a value function that diverges to ∞. The reward measure `M2` specifies the average reward measure for the complete energy consumption which is composed of the continuous components' energy and some restart energy for the components and the system.

4 The LTS semantics of LARES

A $\text{LARES}_{\text{FLAT}}$ model is obtained by a number of transformations applied to standard LARES to construct the instance tree (resolve parameters, Condition, forward and guards statements, cf. [Riedl and Siegle, 2012]) and perform a flattening process from LARES to obtain a planar representation of a system. A $\text{LARES}_{\text{FLAT}}$ model

$$(B, G, M) \in \mathcal{P}(\mathcal{B}) \times multiset(\mathcal{G}) \times \mathcal{P}(\mathcal{M}) \tag{10}$$

comprises a set of instantiated behaviours B, a multiset of guards statements G denoting the interaction between the instantiated behaviours and a set of probability measure statements M. A behaviour instance $b \in B$ can be denoted as a tuple (S, T_U, T_G, s^0), where S is the set of states, T_G is the multiset of guarded transitions (its underlying set is given by $\mathcal{T}_G$), T_U is the multiset of unguarded transitions (with the underlying set $\mathcal{T}_U$) and s^0 denotes the initial state. The universal set of unguarded transitions is a cartesian product between source and target states and the possible distributions $\mathcal{D}$:

$$\mathcal{T}_U = \mathcal{S} \times \mathcal{S} \times \mathcal{D}$$

where a distribution $d \in \mathcal{D}$ is either a (delayed) exponential distribution comprising a rate or an (immediate) discrete distribution comprising a weight. The universal set of guarded transitions is a cartesian product between source and target states and the possible distributions $\mathcal{D}$ but comprises in addition to the unguarded transitions a *guard label* $l \in \mathcal{L}$:

$$\mathcal{T}_G = \mathcal{S} \times \mathcal{S} \times \mathcal{D} \times \mathcal{L}$$

Let the set of behaviour instances B contain $n \in \mathbb{N}$ distinguishable elements $b_1, b_2, \dots, b_n \in B$. Each behaviour instance b_i (with $i \in I^B$, where $I^B := \{1, \dots, n\}$ denotes the index set of the behaviours) is represented by a tuple $(S_i, T_{U\,i}, T_{G\,i}, s_i^0) = b_i$. For two behaviour instances b_i and b_j, where $i \neq j$, the elements are disjoint, i.e. $S_i \cap S_j = \emptyset, T_{U\,i} \cap T_{U\,j} = \emptyset$ and $T_{G\,i} \cap T_{G\,j} = \emptyset$. We define the potential state space $S := S_1 \times \dots \times S_n$ and denote a composed state $s \in S$ as the tuple $(s_1, s_2, \dots, s_n) = s$. For each system of instantiated behaviours the initial composed state $s^0 \in S$ is given by $s^0 := (s_1^0, \dots, s_n^0)$. The goal is to define the semantics of how the system behaves to construct the reachability graph in terms of an LTS.

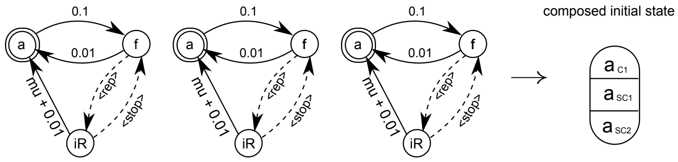

Figure 4 Composed initial state for the running example model

4.1 Semantics: Unguarded Transitions

If one of the instantiated behaviours b_i can perform an unguarded immediate transition $\overset{w}{\dashrightarrow}$ from state s_i to state s_i' then also the composed state s comprising s_i can make the corresponding move. The Structural Operational Semantics (SOS) rule for such an unguarded transition, defined in the style of Plotkin [2004], is:

$$\frac{s_i \overset{w}{\dashrightarrow} s_i'}{(s_1, \ldots, s_i, \ldots, s_n) \overset{w}{\dashrightarrow} (s_1, \ldots, s_i', \ldots, s_n)} \tag{11}$$

A similar SOS rule describes the case for s_i performing an unguarded Markovian transition $\overset{\lambda}{\longrightarrow}$ into the state s_i':

$$\frac{s_i \overset{\lambda}{\longrightarrow} s_i'}{(s_1, \ldots, s_i, \ldots, s_n) \overset{\lambda}{\longrightarrow} (s_1, \ldots, s_i', \ldots, s_n)} \tag{12}$$

For illustration, the example model given in Figure 3 is revisited, for which the composed initial state of the example model (cf. Figure 4) is derived. For brevity, the identifiers of the states are renamed, i.e. `a`, `f` and `iR` instead of `active`, `failed` and `inRep`. As one can see, only a Markovian transition can take place from each of the initial states of the behaviour instances. For determining the states reachable from the composed initial state, the matching SOS rule (12) is applied for each behaviour instance and each unguarded transition leaving the initial state, which yields the next three reachable composed states (cf. Figure 5).

4.2 Semantics: Guarded Transitions

We now provide an informal definition of the semantics for the guarded transitions controlled by guards statements (we do not provide a formal definition of the guarded behaviour semantics in this paper, the interested reader can find it in [Gouberman et al., 2013]).

As depicted by Figure 4, transitions with the guard label ⟨`rep`⟩ may take place from the failed states `f` if triggered by the environment. The events to do so are generated by the guards statement specified in Figure 3 in lines 49..52, which denotes how the behaviour instances interact: as long as `C1` is not failed, the SOS rules for the unguarded transitions can be safely used to further construct the state space (cf. Figure 5). If `C1` and `C2` are failed, the choices `C1.`⟨`rep`⟩ and `C2.`⟨`rep`⟩ as reactions may be triggered, whereas only the reaction `C1.`⟨`rep`⟩ is available if `C2` has not failed.

The container component `C2` is considered to be failed if either subcomponent `SC1` or `SC2` has failed. Whenever the container instance is triggered from the environment via ⟨`rep`⟩, a repair reaction within its behaviour instances is initiated. Due to the delay introduced by the environment, waiting for `C1` to fail before a repair event is generated, it might be the case that

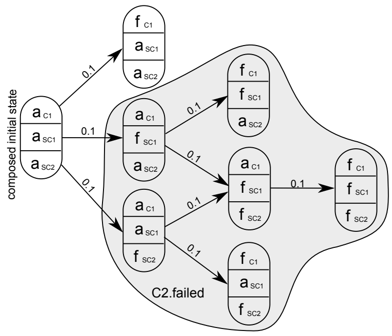

Figure 5 Recurring Application of Rule (12) from the composed initial state for the running example model. Only a subset of all possible unguarded transitions are depicted.

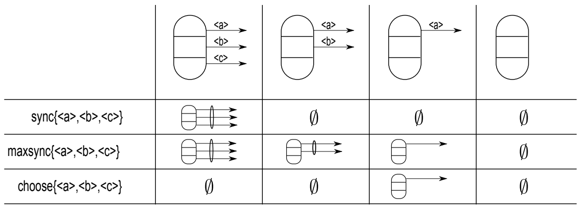

Figure 6 Reactive operator semantics

both subcomponents SC1 or SC2 have already failed. Using the maxsync operator to denote the repair reaction, both subcomponents are repaired at once if needed, or else just one of them. The synchronisation semantics of the operators available to define reactive expressions is illustrated in Figure 6 in a schematic, exemplary fashion. It depicts the cooperation among the behaviour instances from the viewpoint of the current composed state. The operands refer to transition guard labels of the behaviour instances. The content of the table depicts for each operator and the currently available addressed guard labels whether the transitions into the next composed state can be performed simultaneously. As an example from the figure, the choose operator with the operands a, b and c leads to a composed transition, since a minterm $a\bar{b}\bar{c}$ in the disjunctive normal form of the choose operator $a\bar{b}\bar{c} \vee \bar{a}b\bar{c} \vee \bar{a}\bar{b}c$ is fulfilled.

Due to the transformation and flattening process to obtain a $\text{LARES}_{\text{FLAT}}$ model from a user level LARES specification, the guards statement in lines 49..52 is resolved to

```
not C1.B.a guards {
  C1.⟨rep⟩
  maxsync{
    C2.SC1.⟨rep⟩, C2.SC1.⟨rep⟩
  } if (not C2.SC1.B.a) or (not C2.SC2.B.a)
}
```

As an example, in Figure 5 the state $(\mathtt{f}_{\mathtt{C1}}, \mathtt{f}_{\mathtt{SC1}}, \mathtt{f}_{\mathtt{SC2}})$ could be reached. One can see that the generative part of the guards statement is satisfied by this state:

$$(\mathtt{f}_{\mathtt{C1}}, \mathtt{f}_{\mathtt{SC1}}, \mathtt{f}_{\mathtt{SC2}}) \vDash \mathtt{not\ C1.B.a}$$

As a consequence the reactive part has to be considered. Two reactive expressions have been defined, i.e. $\mathtt{C1}.\langle\mathtt{rep}\rangle$ and the conditional reactive comprising the maxsync operator. Since $(\mathtt{f}_{\mathtt{C1}}, \mathtt{f}_{\mathtt{SC1}}, \mathtt{f}_{\mathtt{SC2}})$ satisfies also the by-condition $(\mathtt{not}\ \mathtt{C2.SC1.B.a})$ $\mathtt{or}$ $(\mathtt{not}\ \mathtt{C2.SC2.B.a})$, a choice between both reactions $\mathtt{C1}.\langle\mathtt{rep}\rangle$ and maxsync$\{\mathtt{C2.SC1}.\langle\mathtt{rep}\rangle, \mathtt{C2.SC1}.\langle\mathtt{rep}\rangle\}$ is possible. This case corresponds to the second row first column of Figure 6, i.e. all guard labels referred to by the operands of the maxsync operator are available in the current state $(\mathtt{f}_{\mathtt{C1}}, \mathtt{f}_{\mathtt{SC1}}, \mathtt{f}_{\mathtt{SC2}})$. As a result a composed successor state is $(\mathtt{f}_{\mathtt{C1}}, \mathtt{iR}_{\mathtt{SC1}}, \mathtt{iR}_{\mathtt{SC2}})$ as the addressed transitions with guard label $\langle\mathtt{rep}\rangle$ of $\mathtt{SC1}$ and $\mathtt{SC2}$ are performed synchronously.

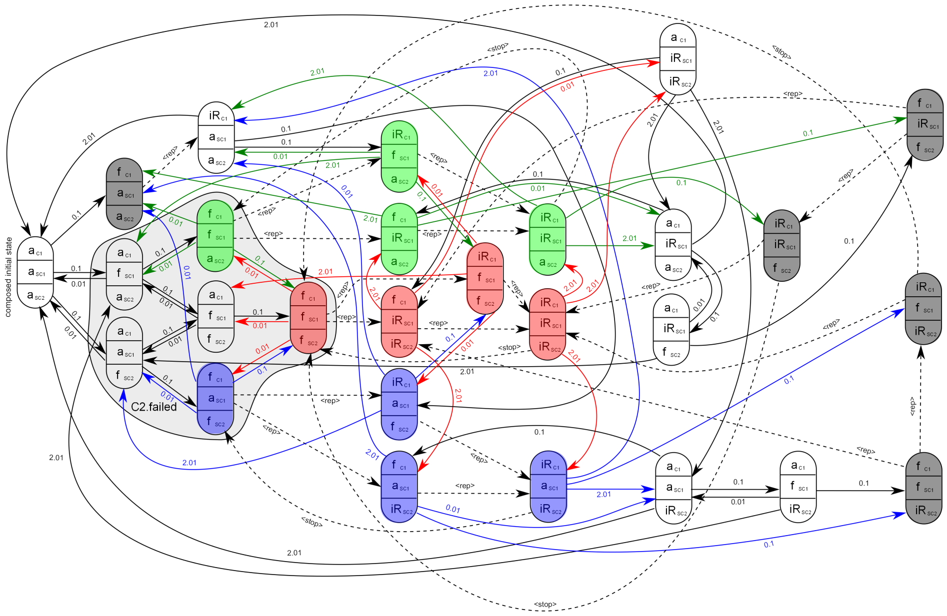

Figure 7 Generated state space for the running example model. It consists of three non-trivial F-ergodic classes (coloured) and five F-transient states (grey). Markovian transitions from F-ergodic classes have the same color as the ergodic class. From F-transient states only immediate transitions are depicted as justified by the maximum progress assumption.

4.3 Performing Reachability Analysis

Taking the initial composed state s^0, the recursive application of the SOS rules (11), (12) and similar rules for synchronised guarded transitions (detailed in [Gouberman et al., 2013]) explores all reachable states as done in Figure 7 for the example model. The generated reachability graph is represented by a transition system $\mathcal{E} = (S, L, \dashrightarrow, \rightarrow, s^0)$ with reachable composed state space S, a set of labels L, weighted immediate transitions $\dashrightarrow \subseteq multiset(S \times S \times \mathbb{R}^+ \times L)$, Markovian transitions $\rightarrow \subseteq multiset(S \times S \times \mathbb{R}^+ \times L)$ and initial state s^0.

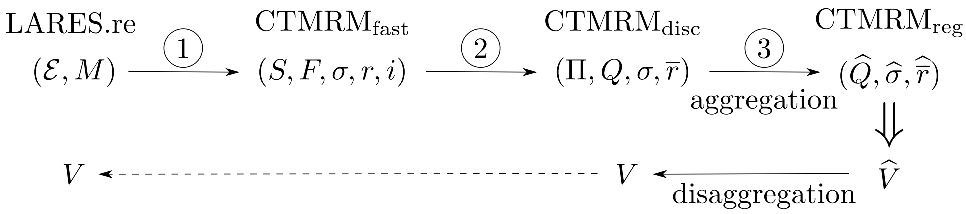

Figure 8 Evaluation of the reward measure M for the flat representation $\mathcal{E}$ of a LARES.re model by a sequence of model transformations.

5 Analysis of LARES Reward Models

In Section 3 we introduced LARES.re as an extension to LARES and described its syntax and informal semantics in Section 4. We now define formal semantics for LARES.re which allows to analyse reward measures. For that purpose we first introduce a planar representation of LARES.re models by extending the LARES$_{\text{FLAT}}$ formalism presented in Section 4. The semantics regarding the reward accumulation process is then defined through model transformation into a CTMRM with fast transitions.

5.1 Planar Representation of LARES.re

The planar representation for LARES.re models builds upon the LARES$_{\text{FLAT}}$ formalism as defined in (10). The set of measures $\mathcal{M}$ comprises reward measures $\mathcal{M}^{RE}$ in addition to the probability measures $\mathcal{M}^{PR}$ of LARES, i.e. $\mathcal{M} := \mathcal{M}^{PR} \cup \mathcal{M}^{RE}$. A reward measure $M \in \mathcal{M}^{RE}$ is a structure $(SE, TE, type)$ where SE is a state reward expression, TE is a transition reward expression and $type$ is the specified reward analysis type, i.e. either total, average or α-discounted (cf. Section 3). The difference between user-level LARES.re and its planar representation is that parameters and hierarchy are resolved, such that all reward expressions consist of atoms which are either numerical values or indicators over conditions which directly point to states of instantiated behaviors.

If one of the state reward or the transition reward expressions is not referenced by the textual specification of the RewardMeasure statement, the associated value will be set to `0`. As an example, the reward measure `M1` defined in Figure 3 by `C1.energy` + `C2.energy discounted 0.01` is resolved to an element $(SE, TE, type) \in \mathcal{M}^{RE}$, where

$$\begin{aligned} SE &= (\texttt{2.5} * [\texttt{C1.B.active}]) + \\ &\quad (\texttt{0.9} * (\texttt{2.5} * [\texttt{C2.SC1.B.active}] + \texttt{2.5} * [\texttt{C2.SC2.B.active}]) + \texttt{0.5}), \qquad (13) \\ TE &= \texttt{0} \text{ and } type = \texttt{discounted 0.01}. \end{aligned}$$

5.2 Transformation to CTMRM with fast transitions

The reachability analysis performed on a LARES$_{\text{FLAT}}$ model as described in Section 4.3 yields a transition system $\mathcal{E} = (S, L, \dashrightarrow, \rightarrow, s^0)$. In this section, we describe the evaluation process for a fixed reward measure $M := (SE, TE, type) \in \mathcal{M}^{RE}$ which results in a value function $V : S \to \mathbb{R}$. This evaluation is based on a sequence of transformations as depicted in Figure 8. In step ① we first transform the pair $(\mathcal{E}, M)$ to a CTMRM with fast transitions $\mathcal{C}_{\text{fast}} := (S, F, \sigma, r, i)$ over the state space S. Hereby we choose a representation of S as integers $\{1, \ldots, n\}$ with $n := |S|$ and regard states as indices (as in Sect. 2.1). The initial distribution is given by $\sigma := \mathbf{1}^T_{s^0} \in \mathcal{D}_n$. The rate reward $r \in \mathbb{R}^n$ is the evaluation of the state reward expression SE on states and the impulse reward $i \in \mathbb{R}^{n \times n}$ is the evaluation of the transition reward expression TE on pairs of states (s, s'), i.e.

$$r_s := eval(SE, s) \quad \text{and} \quad i_{ss'} := eval(TE, (s, s')).$$

If A and B are condition expressions then $eval([A], s) := 1$ if $s \vDash A$ and 0 otherwise and $eval([A \to B], (s, s')) := eval([A], s) \cdot eval([B], s')$. A Markovian transition (s, s', λ, l) is transformed to a slow transition from s to s' with rate λ. Accordingly, an immediate transition (s, s', w, l) is transformed to a fast transition with rate w. Due to the multiset nature of transitions in $\mathcal{E}$, there can be several Markovian and immediate transitions from s to s'. As a consequence of the race condition of exponential distributions these transitions can be merged together by summing up their rates and weights. Thus, we define for all pairs of states (s, s'):

$$S_{ss'} := \sum_{\lambda \in^k \Lambda_{ss'}} k\lambda \quad \text{and} \quad F_{ss'} := \sum_{w \in^k W_{ss'}} kw \quad \text{where}$$

$$\Lambda_{ss'} := \bigcup_{l \in L} \left[\lambda^{(k)} \mid (s, s', \lambda, l) \in^k \rightarrow \right] \quad \text{and} \quad W_{ss'} := \bigcup_{l \in L} \left[w^{(k)} \mid (s, s', w, l) \in^k \dashrightarrow \right]$$

collect all rates and weights with their total multiplicity in a multiset.
A further transformation ② from $\mathcal{C}_{\text{fast}}$ to a discontinuous CTMRM $\mathcal{C}_{\text{disc}} := (\Pi, Q, \sigma, \overline{r})$ and a subsequent reduction ③ by aggregating ergodic classes at zero and eliminating transient states at zero results in a regular CTMRM $\mathcal{C}_{\text{reg}} := (\widehat{Q}, \widehat{\sigma}, \widehat{\overline{r}})$ with $\widehat{Q} := LQR$, $\widehat{\sigma} := L\sigma$ and $\widehat{\overline{r}} := L\overline{r}$. The specified reward measure $type$ is evaluated on $\mathcal{C}_{\text{reg}}$ to the value function $\widehat{V}$ as described in Section 2.1, i.e. $\widehat{V}$ can be the total value $\widehat{V}_\infty$, the α-discounted value $\widehat{V}^\alpha$ or the average value $\widehat{g}$. In order to retrieve the evaluation of $type$ back to the LARES.re model, we perform a disaggregation of the value function $\widehat{V} \in \mathbb{R}^K$ by computing $V := R\widehat{V} \in \mathbb{R}^n$ and consider V as a real-valued function from the set of composed states S.

Note that in LARES.re the transition expression TE is evaluated on a pair of states (s, s') and not on a fixed transition in $\mathcal{E}$. If there are several transitions from s to s' then the impulse reward $i_{ss'}$ is not summed up for each of these transitions but gained only for the case that some of these transitions is performed. This choice of semantics is due to the syntactical definition of a TransitionReward as an expression in a Module definition where the particularly performed transition in the Behavior is not visible at the Module level. This slightly reduces the expressivity of LARES towards greater modularity and is a typical trade-off in the design of modelling languages.

5.3 Implementation issues

The implementation of the transformation of a LARES.re model to a CTMRM with fast transitions can be made more efficient if states s that are not reachable from s^0 are neglected in the CTMRM model. Moreover, pairs of states (s, s') that are not connected by a transition in $\mathcal{E}$ do not have to be considered for the evaluation of impulse rewards. This results in a smaller state space and fewer evaluations of reward expressions. We can also make use of the *maximum progress assumption* which means that Markovian transitions can be neglected

(a,a,a)	(a,a,f)	(a,a,iR)	(a,f,a)	(a,f,f)	(a,f,iR)	(a,iR,a)	(a,iR,f)	(a,iR,iR)
514.05	502.44	513.40	502.44	490.86	501.79	513.40	501.79	512.74
(f,a,a)	(f,a,f)	(f,a,iR)	(f,f,a)	(f,f,f)	(f,f,iR)	(f,iR,a)	(f,iR,f)	(f,iR,iR)
513.68	510.61	510.61	510.61	507.62	507.62	510.61	507.62	507.62
(iR,a,a)	(iR,a,f)	(iR,a,iR)	(iR,f,a)	(iR,f,f)	(iR,f,iR)	(iR,iR,a)	(iR,iR,f)	(iR,iR,iR)
513.68	510.61	510.61	510.61	507.62	507.62	510.61	507.62	507.62

Table 1 Evaluation of reward measure `M1` for the running example model. The composed states are encoded in the order $(\mathtt{C1}, \mathtt{SC1}, \mathtt{SC2})$, e.g. (a,a,a) means $(\mathtt{a}_{\mathtt{C1}}, \mathtt{a}_{\mathtt{SC1}}, \mathtt{a}_{\mathtt{SC2}})$.

in a state s if there is at least one outgoing immediate transition from s. However, due to the underlying semantics by means of a discontinuous CTMRM, the maximum progress assumption is only valid in states that are F-transient. Furthermore, a direct transformation from $(\mathcal{E}, M)$ to the regular CTMRM $\mathcal{C}_{\text{reg}} = (\widehat{Q}, \widehat{\sigma}, \widehat{\overline{r}})$ can be performed without establishing the intermediate models $\mathcal{C}_{\text{fast}}$ and $\mathcal{C}_{\text{disc}}$. All reachable F-transient states can be eliminated and during elimination of such an F-transient state s its total value $(V_\infty^F)_s$ can be computed by a local embedding into a discrete-time Markov reward model together with a geometrical series argument. For the F-ergodic classes that are reached by the eliminated F-transient states the stationary distributions must be computed. Assuming that in practice ergodic classes at zero consist of only a few states, these stationary distributions can be established on the fly during the reachability analysis. During this extended reachability process, a validation of the LARES.re model can be performed by checking for impulse rewards on transitions between states in the same F-ergodic class.

5.4 Analysis of Running Example

In Section 3 we introduced the running example (see Figure 3) for which we now present the evaluation of the specified reward measures `M1` and `M2`. The model consists of 3 instances of Behavior `B`, such that the composed potential state space has 27 states. As one can see in Figure 7 all of these 27 states are reachable from initial state $(\mathtt{a}_{\mathtt{C1}}, \mathtt{a}_{\mathtt{SC1}}, \mathtt{a}_{\mathtt{SC2}})$ and the state space is partitioned into 5 transient states at zero, 10 trivial ergodic classes at zero that consist of a single absorbing state and 3 non-trivial ergodic classes at zero each of them consisting of 4 states. Therefore, the reduced regular CTMRM has 13 states. The α-discounted value function V^α with discount rate $\alpha = 0.01$ is shown in Table 1 and was computed by the evaluation process as outline in Figure 8. Since the reduced CTMRM is ergodic (i.e. has only one recurrent class), the stationary distribution is independent of the initial state, such that P^* has constant rows (8). Therefore, the value function $g = P^*\overline{r}$ for the average reward measure `M2` is constant on S with value $g = 6.57 \cdot \mathbf{1}$.

6 Conclusion and Outlook

We have presented LARES.re as an extension to standard LARES, which allows its users to specify performability measures for dependable, fault-tolerant and dynamically reconfigurable systems. Beside the user-level language, we have defined a planar representation for LARES.re models. The behavioral semantics for standard LARES was originally defined by a model transformation into a state-based LTS formalism, which finally was transformed into a Markov chain. In this paper, we have extended this transformation, by describing a transformation of reward measures into the CTMRM formalism, thereby employing the mathematical concepts of Markov chains with stochastic discontinuities and Markov chains with fast transitions. This finally makes it possible to compute the value functions corresponding to the desired reward measures of interest.

As a next step, we want to formally prove that the semantics for the reward accumulation in form of the finite-horizon total value function V_T as defined in (9) can be also derived as a limit of finite-horizon total value functions V_T^τ with the regular generator Q_τ as $\tau \to \infty$. Furthermore, we want to combine the LARES reward extension with the LARES decision extension (LARES.de) in order to be able to model Markov Decision Processes (MDP) with both Markovian and immediate transitions that lead to continuous-time MDPs with stochastic discontinuities. The reward measures as defined for LARES.re can be used as target functions for optimization criteria. An optimal policy can be computed by solving the non-linear Bellman equations on a reduced regular MDP which maximizes the value function [Guo and Hernandez-Lerma, 2009]. We are also going to complete the implementation of the LARES toolset with respect to both the reward extension and the decision extension (`http://lares.w3.rz.unibw-muenchen.de/`). As future work it is also planned to employ LARES.re and LARES.de in a case study of a real-world critical infrastructure.

Acknowledgements

We would like to thank Deutsche Forschungsgemeinschaft (DFG) who supported this work under grants SI 710/7-1 and for partial support by DFG/NWO Bilateral Research Programme ROCKS.

References

H. Ammar, Y. Huang, and R. Liu. Hierarchical models for systems reliability, maintainability, and availability. *Circuits and Systems, IEEE Transactions on*, 34(6): 629–638, June 1987.

M. Bozzano, A. Cimatti, J.-P. Katoen, V. Y. Nguyen, T. Noll, and M. Roveri. The COMPASS Approach: Correctness, Modelling and Performability of Aerospace Systems. In *Proceedings of the 28th International Conference on Computer Safety, Reliability and Security*, SAFECOMP'09, pages 173–186, 2009a.

M. Bozzano, A. Cimatti, M. Roveri, J.-P. Katoen, V. Y. Nguyen, and T. Noll. Codesign of Dependable Systems: A Component-Based Modeling Language. In *Proceedings of the 7th IEEE/ACM international conference on Formal Methods and Models for Codesign*, MEMOCODE'09, pages 121–130, Piscataway, NJ, USA, 2009b. IEEE Press.

M. Coderch, A. S. Willsky, S. S. Sastry, and D. A. Castanon. Hierarchical aggregation of singularly perturbed finite state Markov processes. *Stochastics*, 8(4):259–289, 1983.

A. Gouberman and M. Siegle. Markov Reward Models and Markov Decision Processes in Discrete and Continuous Time: Performance Evaluation and Optimization. In A. Remke and M. Stoelinga, editors, *Proceedings of the ROCKS Summer School 2012*, Lecture Notes in Computer Science. Springer, 2014.

A. Gouberman, M. Riedl, J. Schuster, M. Siegle, and M. Walter. LARES - A Novel Approach for Describing System Reconfigurability in Dependability Models of Fault-Tolerant Systems. In C. Guedes Soares, R. Bri, and S. Martorell, editors, *Reliability, Risk, and Safety: Theory and Applications. Proceedings of the European Safety and Reliability Conference (ESREL '09)*, volume 1, pages 153–160. CRC Press, 2009.

A. Gouberman, M. Riedl, and M. Siegle. Transformation of LARES performability models to continuous-time Markov reward models. In *Proceedings of the 7th International Workshop on Verification and Evaluation of Computer and Communication Systems (VECOS '13)*, eWiC. British Computer Society, 2013.

X. Guo and O. Hernandez-Lerma. *Continuous-Time Markov Decision Processes: Theory and Applications*, volume 62 of *Stochastic Modelling and Applied Probability*. Springer, 2009.

J.-P. Katoen, I. Zapreev, E. Hahn, H. Hermanns, and D. Jansen. The ins and outs of the probabilistic model checker MRMC. *Performance Evaluation*, 68(2):90–104, 2011.

J. Markovski, A. Sokolova, N. Trčka, and E. Vink. Compositionality for Markov reward chains with fast and silent transitions. *Performance Evaluation*, 66(8):435–452, 2009.

J. Muppala, G. Ciardo, and K. Trivedi. Stochastic reward nets for reliability prediction. *Communications in Reliability, Maintainability and Serviceability*, 1(2):9–20, July 1994.

G. D. Plotkin. A structural approach to operational semantics. *J. Log. Algebr. Program.*, 60-61:17–139, 2004.

G. Point. *AltaRica: Contribution à l'unification des méthodes formelles et de la sûreté de fonctionnement*. Thèse de doctorat, LaBRI – Université Bordeaux I, January 2000.

M. Qureshi, W. Sanders, A. van Moorsel, E. Y, and R. German. Algorithms for the generation of state-level representation of stochastic activity networks with general reward structures. *IEEE Trans. Software Eng.*, 22(9):603–614, 1996.

M. Riedl and M. Siegle. A LAnguage for REconfigurable dependable Systems: Semantics & Dependability Model Transformation. In *Proceedings of the 6th International Workshop on Verification and Evaluation of Computer and Communication Systems (VECOS '12)*, eWiC, pages 78–89. British Computer Society, August 2012.