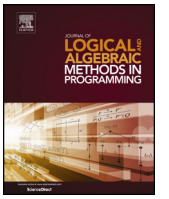
Journal of Logical and Algebraic Methods in Programming 101 (2018) 110–131

Contents lists available at ScienceDirect

Journal of Logical and Algebraic Methods in Programming

www.elsevier.com/locate/jlamp



Probabilistic timed graph transformation systems

Maria Maximova [a,*], Holger Giese [a], Christian Krause [b]

[a] Hasso Plattner Institute at the University of Potsdam, Germany
[b] SAP SE, Potsdam, Germany

ARTICLE INFO

Article history:
Received 31 December 2017
Received in revised form 27 July 2018
Accepted 7 September 2018
Available online xxxx

Keywords:
Graph transformations
Probabilistic timed automata
PTCTL
PRISM model checker
HENSHIN

ABSTRACT

Today, software has become an intrinsic part of complex distributed embedded real-time systems. The next generation of embedded real-time systems will interconnect the today unconnected systems via complex software parts and the service-oriented paradigm. Due to these interconnections, the architecture of systems can be subject to changes at run-time, e.g. when dynamic binding of service end-points is employed or complex collaborations are established dynamically. However, suitable formalisms and techniques that allow for modeling and analysis of timed and probabilistic behavior of such systems as well as of their structure dynamics do not exist so far.

To fill the identified gap, we propose Probabilistic Timed Graph Transformation Systems (PTGTSs) as a high-level description language that supports all the necessary aspects of structure dynamics, timed behavior, and probabilistic behavior. We introduce the formal model of PTGTSs in this paper as well as present and formally verify a mapping of models with finite state spaces to probabilistic timed automata (PTA) that allows to use the PRISM model checker to analyze PTGTS models with respect to PTCTL properties.

1. Introduction

Today, software has become an intrinsic part of complex distributed embedded real-time systems, which need to realize more advanced functionality. The next generation of embedded real-time systems will interconnect the today unconnected systems via complex software parts and the service-oriented paradigm. It is envisioned that such networked systems will be able to behave much more intelligently by building communities of autonomous agents that exploit local and global networking to adapt and optimize their functionality [1].

In contrast to today's real-time systems, their behavior will be additionally characterized by *structure dynamics* that results from their complex coordination behavior. This structure dynamics requires execution in real-time and reconfiguration at run-time to adjust the systems behavior to its changing context and goals, leading to self-adaptation and self-optimization [2]. For these systems, also the structure resp. architecture is subject to changes at run-time, e.g. when dynamic binding of service end-points is employed or complex collaborations are established dynamically. In the latter case, often the structural context in the form of local topology and distribution information is particularly important.

As a concrete example for such an advanced embedded real-time system, the RailCab research project [3] aims at combining a passive track system with intelligent shuttles that operate autonomously, act individually, and make independent and decentralized operational decisions. For the RailCab application example it holds that some functionality may be safety-

* Corresponding author.
E-mail addresses: maria.maximova@hpi.de (M. Maximova), holger.giese@hpi.de (H. Giese), christian.krause01@sap.com (C. Krause).

https://doi.org/10.1016/j.jlamp.2018.09.003
2352-2208/

critical such as the convoy coordination, or mission-critical for economic reasons such as the negotiation of the transport contracts. Furthermore, the required properties are not merely qualitative ones but also quantitative ones involving time as well as probabilities. For instance, convoy coordination protocols have to be established between shuttles nearby in the topology, usually involving hard real-time constraints, and the sent protocol message may be lost with a non-zero probability. Consequently, we need methods and tools to guarantee critical quantitative properties when developing such systems, which include *structure dynamics*, *timed behavior*, and *probabilistic behavior*.

Combinations of different modeling approaches have led to a number of new interesting applications in the last couple of years. In the following, we briefly describe related modeling and analysis approaches, which combine some of the aspects of *structure dynamics*, *timed behavior*, and *probabilistic behavior*.

Timed graph transformation systems (TGTSs) [4–6] facilitate the modeling of timed behavior in graph transformation systems using timed automata concepts.[1] Specifically, nodes can be annotated with real-valued clocks which can be dynamically added and removed from the systems. Rules can include clock constraints as additional application conditions, and clocks can be reset. Using symbolic, *zone*-based representations [8,9] and an implementation in an extension [5] of the GROOVE tool [10], the state spaces of TGTSs can be explored and analyzed, e.g. for time-bounded reachability checks. Moreover, inductive invariant checking [6] for TGTSs provides a means to deal with infinite-state systems. Thus, TGTSs enable the analysis of combined models with structure dynamics and real-time behavior. However, probabilistic behavior is not supported.

A combination of structure dynamics and probabilistic behavior is supported by probabilistic graph transformation systems (PGTSs) [11], which are an extension of the graph transformation theory with discrete probabilistic behavior. In PGTSs, transformation rules are allowed to have multiple right-hand sides, where each of them is annotated with a probability. The choice for a rule match is nondeterministic, whereas the effect of a rule is probabilistic. This approach can be used to model randomized behavior and on-demand probabilistic failures, such as message loss in unreliable communication channels and supports modeling and analysis by an extension of the HENSHIN [12] tool and a mapping to the PRISM [13] model checker.[2]

Real-time rewrite theories as supported by the executable specification language of Real-Time Maude [15] facilitate combined modeling of structure dynamics and real-time behavior. Analysis goals include reachability checks for failures of safety properties and model checking of time-bounded temporal logic properties. Such properties are in general not decidable and therefore the provided tool support is incomplete.

Probabilistic rewrite theories implemented in PMaude [16] provide a combination of structure dynamics, probabilistic behavior for discrete branching, and stochastic timed behavior. Properties for PRTs are specified using probabilistic temporal logic and checked using discrete event simulation, e.g. using the Vesta tool [17]. However, in order to simulate and analyze models in PMaude, *all* nondeterminism has to be resolved, i.e., neither discrete nondeterministic choice nor timed nondeterminism as required for real-time behavior, are allowed.

Probabilistic Timed Automata (PTA) [18] combine the modeling features of Markov decision processes (MDPs) [19] and timed automata (TA) [20,21] and thereby allow to analyze systems exhibiting both timed and probabilistic phenomena. Analysis goals for PTA include the checking of probabilistic time-bounded reachability, computation of rewards, as well as PTCTL model checking [18]. Such properties can be analyzed for PTA, e.g., using the PRISM tool.

The timed and probabilistic extensions of rewrite systems, specifically rewrite theories in Maude variants and GTSs, provide the best coverage for the required modeling features. However, none of the existing models facilitates the modeling and analysis of *all* identified requirements.

To fill the identified gap, we propose to combine and extend the existing models to the formalism of Probabilistic Timed Graph Transformation Systems (PTGTSs) that supports modeling and analysis of *structure dynamics*, *timed behavior*, and *probabilistic behavior*. We introduce the formal model of PTGTSs in this paper and present a formally verified mapping of models with finite state spaces to probabilistic timed automata (PTA) that allows to use the PRISM model checker to analyze PTGTS models with respect to PTCTL properties.[3]

This paper is structured as follows. First, the necessary prerequisites in form of probabilistic timed automata (PTA) are recapitulated in Section 2. Then, we introduce Probabilistic Timed Graph Transformation Systems (PTGTSs) in Section 3. In Section 4, we define a mapping of PTGTSs into the corresponding PTA and show that this mapping is sound with respect to the corresponding semantics. Subsequently in Section 5, we present the tool support for our approach using the graph transformation tool HENSHIN and apply it to model our running example handling a shuttle scenario. Finally in Section 6, we consider the analysis of PTGTS models by combining the state space generation of HENSHIN and the PTA model checking of the PRISM tool via a mapping. The paper is closed with some final conclusions and an outlook on planned future work.

[1] An alternative approach for graph transformation systems with time was developed in [7]. However, this approach is not suitable in our context since symbolic state space representations and quantitative analysis methods are not considered in [7].

[2] Also stochastic graph transformation systems (SGTSs) [14] that incorporate stochastic timed behavior into GTSs by including continuous-time probability distributions that describe the average delay of firing of rules, once they are enabled, have been proposed. However, note that they do neither support probabilistic behavior nor real-time behavior as they assume a different model of time.

[3] This is an extended version of [22] that includes in addition the before omitted formal construction and verification of the mapping of probabilistic timed graph transformation systems to probabilistic timed automata (Section 4), two further examples demonstrating the contributions and present our main running example in all detail, further definitions to provide explanations at a more precise formal level, and further visualizations of the results and more detailed explanations for the exemplary analysis of our running example using our tool chain.