Theoretical Computer Science 594 (2015) 24–43

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Checking conformance for time-constrained scenario-based specifications ☆

S. Akshay [a,*], Paul Gastin [b], Madhavan Mukund [c], K. Narayan Kumar [c]

[a] Indian Institute of Technology Bombay, India
[b] LSV, ENS Cachan, INRIA, CNRS, France
[c] Chennai Mathematical Institute, Chennai, India

ARTICLE INFO

Article history:
Received 4 March 2014
Received in revised form 1 February 2015
Accepted 18 March 2015
Available online 24 March 2015
Communicated by P. Aziz Abdulla

Keywords:
MSC graphs
Timed automata
Model checking

ABSTRACT

We consider the problem of model checking message-passing systems with real-time requirements. As behavioral specifications, we use message sequence charts (MSCs) annotated with timing constraints. Our system model is a network of communicating finite state machines with local clocks, whose global behavior can be regarded as a timed automaton. Our goal is to verify that all timed behaviors exhibited by the system conform to the timing constraints imposed by the specification. In general, this corresponds to checking inclusion for timed languages, which is an undecidable problem even for timed regular languages. However, we show that we can translate regular collections of time-constrained MSCs into a special class of event-clock automata that can be determinized and complemented, thus permitting an algorithmic solution to the model checking/conformance problem.

1. Introduction

In a distributed system, several agents interact to generate a global behavior. This interaction is usually specified in terms of scenarios, using message sequence charts (MSCs) [23]. Protocol specifications typically include timing requirements for messages and descriptions of how to recover from timeouts, so a natural and useful extension to MSCs is to add timing constraints between pairs of events, yielding time-constrained MSCs (TCMSCs) [8,1].

Infinite collections of MSCs are typically described using message sequence graphs (MSGs) [23,9]. An MSG, a finite directed graph with nodes labeled by MSCs, is the most basic form of a High-level Message Sequence Chart (HMSC) [26]. Any path through the graph generates a new MSC by concatenating the MSCs seen along the path. Thus, the set of all paths through an MSG generates a possibly infinite collection of MSCs. In this article, we generalize MSGs to time-constrained MSGs (TCMSGs), where nodes are labeled by TCMSCs and edges may have additional time constraints between nodes. Thus, TCMSGs generate infinite collections of time-constrained scenarios, i.e., TCMSCs. This forms our basic model of specification.

A natural system model in this setting is a timed message-passing automaton (timed MPA), a set of communicating finite-state machines equipped with clocks that are used to guard transitions, as in timed automata [11]. Just as timed words are used to describe the runs of timed automata, the interactions exhibited by timed MPAs can be described using

☆ Supported by CNRS, LIA InForMel and DST-INSPIRE faculty award [IFA12-MA-17].
* Correspondence to: Dept. of Computer Science and Engineering, IIT Bombay, Powai, Mumbai, 400076, India. Tel.: +91 2225767711.
E-mail addresses: akshayss@cse.iitb.ac.in (S. Akshay), Paul.Gastin@lsv.ens-cachan.fr (P. Gastin), madhavan@cmi.ac.in (M. Mukund), kumar@cmi.ac.in (K. Narayan Kumar).

http://dx.doi.org/10.1016/j.tcs.2015.03.030
0304-3975/

timed MSCs—MSCs in which each event is assigned an explicit timestamp. However, the global state space of a timed MPA in fact defines a timed automaton over a distributed alphabet and in this paper we focus on this simplified global view of timed message-passing systems, though our results go through smoothly for the distributed system model as well. Thus, our main interest in this paper lies in considering a distributed specification (formalized using TCMSGs) and comparing it against a global timed implementation.

Our aim is to check if all timed MSCs accepted by a timed MPA conform to the time constraints given by a TCMSG specification. This problem can naturally be seen as comprising of two parts. The first asks if for a given timed MPA A and TCMSG G, every timed execution exhibited by A is in the specification. Indeed, this is the standard model-checking question for timed MPAs. The second part, the coverage problem, asks if every TCMSC generated by a given TCMSG can be witnessed by some timed execution of the TMPA. To make the problem tractable, we focus on *locally synchronized* TCMSGs—those for which the underlying behavior is guaranteed to be regular [22].

In general, the model checking problem above corresponds to checking inclusion for timed languages, which is known to be undecidable even for timed regular languages [6]. Fortunately, it turns out that timing constraints in a TCMSG correspond to a very restricted use of clocks. This allows us to associate with each TCMSG an extended event clock automaton that accepts all timed executions that are consistent with the timing constraints of the TCMSG. We prove that these extended event clock automata can be determinized and complemented (as in the case of ordinary event clock automata [7]), yielding an algorithmic solution to our model checking problem.

Turning to the coverage problem, we observe this cannot be directly reduced to a timed inclusion problem. The timed inclusion problem in this direction would ask if there is a witnessing execution of the timed MPA for every timed linearization of a TCMSC generated by the TCMSG. But an implementation (timed MPA) having strictly better time bounds than the specification might have a witnessing execution for every TCMSC generated by the TCMSG, even if it does not satisfy every timed linearization of the TCMSC. Such an implementation should be considered as a valid one and this is precisely what our definition of the coverage problem achieves. For solving this problem, we need an additional assumption on the specification. We assume that the locally synchronized TCMSG has a special form that every process on the TCMSC labeling any node has some event. Now, we use the same extended event clock automaton as above accepting all timed executions that are consistent with the TCMSG. Then, using a product construction, we can recover the set of paths of the TCMSG which have some valid execution in the timed MPA, thus solving the coverage problem.

Related work. We have used TCMSGs as the basic model for specifying high level distributed and timed systems. However, there are other formalisms which also tackle time and concurrency issues in systems. In Petri nets [30] *tokens* are positioned in *places* and a transition fires by consuming tokens and creates new ones, in general in other places. Thus, transitions that consume different tokens, can fire independently. Many timed extensions of Petri nets have been considered, for instance, time Petri nets [12], timed Petri nets [29]. Unfoldings of Petri nets provide a way to model the partial order behavior of these systems and by lifting these unfoldings to the timed extensions, they provide a timed partial order semantics [17]. For more discussion on this refer to [16]. However, these unfoldings are seldom graphically representable in a compact manner unlike MSCs (and their timed extensions). Further, unfoldings in Petri nets correspond to "branching time" whereas MSCs express "linear time" behavior.

Other models dealing with time and concurrency include networks of timed automata [6] and products of timed automata [20]. Again in [13], unfolding techniques were applied to study such networks of timed automata. However, these models do not allow communication via explicit message passing which is one of the main features of the timed MPA and TCMSGs that we have introduced.

The formal semantics and analysis of timing in MSCs has been addressed earlier in [8,10,15,24]. In [8] and [10], only single timed MSCs or high-level timed MSCs were considered, while in [24] one of the first models of timed MPAs was introduced. However, the latter do not consider MSCs as a semantics of their automata but rather look at restricted channel architectures (e.g., one-channel systems) to transfer decidability of reachability problems from the untimed to the timed setting. The automaton model in [15] links the two approaches by considering a similar automaton model with semantics in terms of timed MSCs. But they tackle only a specific matching problem for which they propose a practical solution using the tool Uppaal. More recently, in [4] the authors have considered TCMSGs under restrictions that are weaker than being locally-synchronized. Though this allows modeling more general non-regular languages of TCMSCs, they only tackle the emptiness problem and do not address more complicated issues of consistency or conformance as we do.

In [19], the authors develop a specification theory that combines notions of specifications and implementations and provides constructs for checking consistency etc., in the setting of sequential real-timed systems. However, they define the implementation as another specification and relate the two using a notion of refinement defined as an alternating (timed) simulation relation. In our setting, the implementation and specification are different objects to begin with and we relate them at the level of behaviors rather than systems. Thus, checking consistency corresponds to checking inclusion of timed behaviors which is often a harder problem than defining a simulation. In addition, we consider timed and distributed systems, where concurrency plays a major role and gives rise to several additional challenges.

Preliminary versions of some of the results were presented as extended abstracts in [3,5]. Here, we establish a generic framework that combines those results as well as completes and generalizes the proofs and techniques.

Structure of the paper. The paper is organized as follows. We begin with some preliminaries where we introduce (timed) MSCs, MSGs and the timed automata formalisms. In the subsequent section, we discuss the conformance problem in detail.