Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Passive testing of communicating systems with timeouts

Mercedes G. Merayo [*], Alberto Núñez

Departamento de Sistemas Informáticos y Computación, Universidad Complutense de Madrid, 28040 Madrid, Spain

ARTICLE INFO

Article history:
Received 8 April 2014
Received in revised form 24 March 2015
Accepted 27 March 2015
Available online 8 April 2015

Keywords:
Software testing
Communicating systems
Formal methods
Timeouts
Passive testing

ABSTRACT

Context: The design of complex systems demands methodologies to analyze its correct behaviour. It is usual that a correct behaviour is determined by the compliance with temporal requirements. Currently, testing is the most used technology to validate the correctness of systems. Although several techniques that take into account time aspects have been proposed, most of them require the tester interacts with the system. However, if this is not possible, it is necessary to apply a passive testing approach where the tester monitors the behaviour of the system.
Objective: The aim of this paper is to propose a methodology to perform passive testing on communicating systems in which the behaviour of their components must fulfill temporal restrictions associated with both performance and delays/timeouts.
Method: Our framework uses algorithms for checking traces collected from the systems against invariants which formally represent the most relevant properties that must be fulfilled by the system. In order to support the feasibility of the methodology, we have performed an empirical study on a complex system for automatic recognition of images based on a pipeline architecture. We have analyzed the correctness of the system's behaviour with respect to a set of invariants. Finally, an experiment, based on mutations of the system, was conducted to study the level of detection of a set of invariants.
Results: Different errors were detected and fixed along the development of the system by means of the proposed methodology. The results of the experiments with the mutated versions of the system indicated that the designed set of invariants was more effective in finding errors associated to temporal aspects than those related to communication among components.
Conclusion: The proposed technique has been shown to be very useful for analyzing complex timed systems, and find errors when the tester has no control over their behaviour.

1. Introduction

Real-time systems are a kind of systems whose behaviour depends on timed conditions. Thus, the necessity of checking and determining their correctness requires the application of a methodology which considers time requirements. In order to perform this task, several techniques, algorithms, and semantic frameworks have been introduced in the literature. Among them, *testing* is probably the most widely used in industrial environments. Even though testing was classically seen as an *informal* discipline, several studies have tried to reconcile testing and formal methods [11,17]. In testing, there is usually a distinction between two approaches: *Passive* and *Active*. The main difference between them is whether a tester can interact with the implementation under test. If the tester can interact with the tested system, then we are in the active testing paradigm. On the contrary, if the tester simply monitors the behaviour of the system, then we are in the passive testing paradigm. During the last years, the testing community has shown a growing interest to take into account time aspects. Most of the proposals use an active testing approach [39,22,38,16,32,31,42,28]. Nevertheless, some works on formal passive testing of timed systems [2,3] have been proposed recently. In most models, temporal requirements usually refer to the time that a system consumes for performing operations. However, another kind of temporal conditions can affect systems: *Timeouts*. A timeout is a specified period of time that will be allowed to elapse in a system waiting for an interaction with the environment; if the period ends and no action has been produced, the internal state of the system changes and its reactions to the actions that are received from the environment may be different. The notion of timeout has been included in different active testing frameworks [7,5,21,31]. However, it is very frequent that the tester is unable to interact with the implementation under test, due to the fact that the system must be running without interruption. In these cases, it is required to

* Corresponding author.
E-mail addresses: mgmerayo@fdi.ucm.es (M.G. Merayo), alberto.nunez@pdi.ucm.es (A. Núñez).

http://dx.doi.org/10.1016/j.infsof.2015.03.009
0950-5849/© 2015 Elsevier B.V. All rights reserved.

apply a passive testing approach such that the tester only monitors the behaviour of the system, without affecting its behaviour. In a previous work [27], a formal methodology to analyze this kind of systems is provided. Taking as initial step, on the one hand, our model to specify systems that present timeouts [31] and, on the other hand, our work on passive testing of timed systems [3], a new passive testing methodology [27] has been proposed for systems that present restrictions over both the time associated with actions and timeouts. In order to model this kind of systems, we use a formalism based on *finite state machines* since it allows us to specify in a natural way both kinds of temporal conditions. The goal of the proposed framework is to provide a technique to check whether the execution of a system violates a given *invariant* which formally expresses a property that must be fulfilled. If an error is detected when a log extracted from the system under test is checked against an invariant, then it can be claimed that the system does not conform to the specification. A novel type of invariant, so called *to-invariants*, was defined in order to represent properties that take into account the different behaviours of the system induced by its possible timeouts. It is worth to note that possible timeouts can strongly influence the actions that the system can perform after they have been triggered. Therefore, timeouts may affect the functional behaviour of the systems. This *influence* is hidden to the external observer since we can consider timeouts as a kind of internal action: we do not directly know either when they are triggered or the state reached after they are triggered. In addition to present the syntax and examples of the new type of invariant, algorithms to check both the correctness of to-invariants with respect to a specification and the correctness of the logs with respect to to-invariants are given. The soundness of the methods for this kind of invariants was shown by relating them to a conformance relation according to the interpretation of good implementation for a given specification.

Although many systems can be modelled by a single finite state machine, the growing necessity to design complex systems that present interconnected components requires the development of frameworks that allow to analyze the correctness of the communication among them. This paper represents an extension of the proposed methodology [27] in order to deal with properties related to the communication actions. An adaptation of *communicating finite state machines* formalism is used to represent this kind of systems, where each of the components in the net are given by a timed finite state machine. A new kind of invariant, so called *communicating invariants*, is introduced to allow the representation of properties that involve the communication among the components. The corresponding algorithms to check the correctness of the invariants and the logs are also provided.

An additional contribution of this paper is a complete case study where a non-trivial pipeline of image processing is used. The system proposed in this study performs the automatic recognition of a set of images using a pipeline model for parallel processing. In this approach, a set of X-ray images of human skulls are processed to automatically detect the frontal sinus [41]. The main functionalities of the system have been modelled by timed finite state machines that can evolve by raising timeouts and can include temporal constraints over the action durations. A set of to-invariants and communicating invariants are designed and used to check the correctness of a set of traces collected from the system. The system has been implemented using SIMCAN, a simulation platform aimed to model parallel and distributed systems [35]. This implementation has been carried out in three different phases. First, the underlying hardware architecture of a complete distributed system has been modelled. Second, the application used for the automatic recognition of images has been written using the API provided by SIMCAN. The last phase corresponds to the execution of the implementation in the simulated environment to collect traces that could be used to analyze the correctness of the system's behaviour. In order to apply our approach, we extended the PASsive TEsting tool (PASTE [3]), which allows users to work with the formal framework presented in this paper. In particular, all the algorithms for checking the correctness of the traces and the invariants have been implemented. Then, they have been applied to a set of invariants and a set of traces obtained from the implementation in order to support the feasibility of the proposed methodology. Finally, we have studied the performance of the invariants applying a method based on mutation testing techniques [37,40,34]. The implementation was *mutated* in order to inject errors. Each of the obtained mutants were executed and the traces collected were checked against the set of available invariants in order to determine, based on the obtained results, their level of fault detection. The idea is that if an invariant finds many errors in the traces obtained from the execution of mutants, then this invariant is more likely to find an error in a faulty implementation.

This paper extends and enhances our previous work [27]. Specifically, we can mention the following contributions.

- The new passive testing framework to deal with communicating systems is original. An *adaptation* of communicating finite state machines formalism is introduced to represent this kind of systems, where each of the components in the net are given by a timed finite state machine. A new kind of invariant is proposed to allow the representation of properties that involve the communication among the components. The corresponding algorithms to check the correctness of the invariants with respect the specification and the logs against the invariants, are also provided. This is a non-trivial extension and the considered systems add substantial expressive power to the framework.
- We have extended the passive testing tool, PASTE, which allows users to work with the formal framework introduced in this paper.
- A complete case study, based on an image processing system, is presented. The different components of the system have been modelled by timed finite state machines and a set of to-invariants and communicating invariants have been designed. The system has been implemented and executed in a simulated environment, SIMCAN, in order to support the feasibility of the proposed methodology. Specifically, we have studied the performance of the invariants applying a method based on mutation testing techniques.

The rest of the paper is structured as follows. Section 2 provides preliminary material. Section 3 reviews our previous work on passive testing for timed systems with timeous. We present the formalism to specify timed systems, the description of to-invariants and the algorithms related to the decision of their correctness with respect to a specification and the correctness of observed logs with respect to to-invariants. This section also contains the material related to the correctness of the approach. Section 4 contains our passive testing framework for communicating systems. Section 5 presents the empirical study of the image processing system and the results obtained from the experiments. Section 6 identifies possible threats to validity of the experimentation phase. Section 7 reviews the main proposals to model systems that present timeouts and to represent communication among different components in a system. We finish this paper, in Section 8, with our conclusions.

2. Notation

Given a set A, we let A^* denote the set of finite sequences of elements of A; $\epsilon \in A^*$ denotes the empty sequence. Given a sequence $e \in A^*, e_r \in A$, with $1 \leqslant r \leqslant |e|$, denotes the r-th element of e. Finally, let $e \in A^*$ and $a \in A$. We have that $e \cdot a$ denotes the sequence e followed by a and $a \cdot e$ denotes the sequence e preceded by a.