Science of Computer Programming 126 (2016) 52–72

Contents lists available at ScienceDirect

Science of Computer Programming

www.elsevier.com/locate/scico

A family of test selection criteria for Timed Input-Output Symbolic Transition System models

Alan Moraes [a,b,*], Wilkerson L. Andrade [b], Patrícia D.L. Machado [b]

[a] Informatics Center, Federal University of Paraíba, João Pessoa, PB, Brazil
[b] Software Practices Laboratory, Federal University of Campina Grande, Campina Grande, PB, Brazil

ARTICLE INFO

Article history:
Received 24 April 2015
Received in revised form 6 May 2016
Accepted 9 May 2016
Available online 12 May 2016

Keywords:
Software testing
Model-based testing
Test selection criteria
Test adequacy criteria
Real-time systems

ABSTRACT

Test selection criteria can be applied to select, from a test suite, cost-effective subsets that are both cheaper to execute and as effective at detecting faults as the original suite. However, the choice of a test selection criterion is not straightforward for real-time systems, because most criteria presented in literature are for untimed systems and there is a lack of studies that investigate their cost-effectiveness when compared to specific criteria for real-time systems. In this paper, we investigate the cost-effectiveness of test selection criteria in test suite reduction for model-based testing of real-time systems, in particular for Timed Input-Output Symbolic Transition Systems (TIOSTS) models. First, we defined 18 test selection criteria for TIOSTS models and formalized a hierarchy of criteria partially ordered by strict inclusion. The defined criteria include transition-based criteria, data-flow-oriented criteria and real-time systems criteria. Second, we evaluated the cost-effectiveness of the criteria in an experiment. In the empirical study, we used a TIOSTS specification of a burglar alarm system, an implementation that simulated the system, and faults seeded into the implementation by using mutation testing. Results showed that, despite being a criterion for untimed systems, ALL-TRANSITION-PAIRS was the most cost-effective criterion for test case selection in model-based testing of real-time systems. We conclude that more cost-effective criteria that explore time-related features of models of real-time systems are still needed.

1. Introduction

Model-Based Testing is a testing approach that relies on the existence of abstract models of an application to generate, execute and evaluate tests [1]. It has been applied with success in industry, with special emphasis in avionic, railway, and automotive domains [2,3]. In model-based testing, test case generation has an important role. It consists of a systematic search for test cases that can be extracted from models. This search is usually guided by *test selection criteria*[1] which express testing objectives as well as stop conditions. These criteria define which parts of the model are going to be tested, how often and under what circumstances. Additionally, different test case generation algorithms can be applied. As a result, automatically generated test suites may vary in size, cost, behavior coverage, and fault detection capability, depending on the choice of criteria and algorithms.

* Corresponding author at: Informatics Center, Federal University of Paraíba, João Pessoa, PB, Brazil.
E-mail addresses: alan@ci.ufpb.br (A. Moraes), wilkerson@computacao.ufcg.edu.br (W.L. Andrade), patricia@computacao.ufcg.edu.br (P.D.L. Machado).

[1] Test selection criteria have many synonyms in literature: test data adequacy criteria, test adequacy criteria, test coverage criteria, or just test criteria.

http://dx.doi.org/10.1016/j.scico.2016.05.004

Usually, generated test suites are large and potentially expensive to manage and execute. Therefore, test case selection techniques can be applied post-generation to obtain a smaller test suite that meets a certain requirement, for example, a given number of test cases [4]. Moreover, test suite reduction techniques are concerned with selecting a subset of test cases that also covers the same test selection criterion as the original test suite [5], whereas test case prioritization techniques define an execution order of test cases according to a given testing goal, particularly detecting failures as early as possible [6]. Here, we focus on test suite reduction.

On the other hand, real-time systems are reactive systems whose behavior is constrained by time [7]. Therefore, the testing of these systems should also uncover time-related faults which may require specific test cases to be exercised. Most test selection criteria for models of real-time systems are based on coverage of structural elements of the model behavior and its data usage [8]. Some specific test selection criteria for real-time systems have been proposed, such as covering all clock resets [9]. However, the choice of a criterion is not straightforward, because most criteria presented in literature are for untimed systems and there is a lack of studies that investigate their cost-effectiveness when compared to specific criteria for real-time systems [10].

In this paper, we investigate the cost-effectiveness of test selection criteria for Timed Input-Output Symbolic Transition Systems (TIOSTS) models [11,12] in the test suite reduction problem. Real-time system behavior is modeled in TIOSTS models as a transition system with data and time symbolically defined. Transition systems such as TIOSTS models are often used for conformance testing of real-time systems [13,14].

Here, we make two contributions. The first contribution is the definition of 18 test selection criteria for TIOSTS models and the formalization of a hierarchy of criteria partially ordered by strict inclusion [15]. The defined criteria include untimed transition-based and data-flow-oriented criteria, and specific criteria for real-time systems. The second contribution is a controlled experiment that compares the cost-effectiveness of the criteria in test suite reduction for model-based testing of real-time systems. In the empirical study, we used a TIOSTS specification of a burglar alarm system, an implementation that simulated the system, and faults seeded into the implementation by using mutation testing. Results have shown that, despite being a criterion for untimed systems, the criterion All-Transition-Pairs was the most cost-effective in the experiment.

This paper is an extension of our prior work [10]. We extended it in the following ways. First, we refined the definitions of the criteria. Second, we conducted a new empirical study. The evaluation of our prior work was preliminary and no implementation of the system under test was considered. Here, our evaluation is closer to the industrial setting, because we implemented a simulation of the system under test, created faulty implementations of the system under test, generated tests targeting the system, and executed them against the faulty implementations to compare cost-effectiveness of the criteria. Third, we improved the statistical treatment of our analysis and included a qualitative assessment of the results.

The paper is structured as follows. Section 2 introduces the TIOSTS model which we use in this work. Section 3 defines the criteria for TIOSTS models and formalizes a hierarchy with them. Section 4 presents an empirical study of test suite reduction to compare the criteria. Section 5 discusses the findings of this work and relates them to results in literature. Finally, Section 6 presents concluding remarks.

2. Timed Input-Output Symbolic Transition System model

Timed Input-Output Symbolic Transition System (TIOSTS) [11,12] is a symbolic model for real-time systems that handles both data and time. The TIOSTS model was defined as an extension of two existing models: Timed Automata [16] and Input-Output Symbolic Transition Systems [17,18]. Basically, a TIOSTS is an automaton with a finite set of locations where system data and time evolution are respectively represented by variables and a finite set of clocks. The transitions of the model are composed of a source location, a target location, a guard on variables and clocks, an action with parameters, an assignment to variables, and a set of clocks to reset.

Fig. 1 shows an example of TIOSTS that models a machine for refilling a card for using the subway.[2] Initially, the system is in the `Idle` location where it expects the `Credit` input carrying the desired `value` to refill, then this value is saved into the `refillValue` variable[3] and `balance` is initialized to zero.

From the `Receive` location to `Verify`, the client informs the amount to be credited to the card. This value is accumulated in the `balance` variable and the `clock` is set to zero. If the current balance is less than the desired value to refill, then the `Receive` location is reached again and the `MissingValue` output is emitted for informing the remaining value (the condition `value = refillValue − balance` contained in the guard means “choose a value for the `value` parameter that, with the values of `refillValue` and `balance` variables, satisfies the guard”).

From the `Verify` location, if the `balance` is greater than `refillValue` some value must be returned to the client in less than 5 time units. After that, the `clock` is reset to zero again. Then, the `RefillCard` output action must be performed in less than 5 time units and the `cardBalance` is increased by `refillValue`. Otherwise, from `Verify`, if `balance` is exactly equal to `refillValue`, then the card must be refilled in less than 5 time units. Finally, from the `Print` location, the voucher must be printed in less than 15 time units and `Idle` location is reached again. The formal definition of TIOSTS models is presented in Definition 1 [11].

[2] In graphical representations of TIOSTS models, the pipe symbol is used to separate multiple assignments and clock resets in the same transition. Also, multiple assignments are executed from left to right.

[3] Action parameters have local scope, thus their values must be stored in variables for future references.