Science of Computer Programming ••• (••••) •••–•••

Contents lists available at ScienceDirect

Science of Computer Programming

www.elsevier.com/locate/scico

Model-based testing for building reliable realtime interactive music systems [☆]

Clement Poncelet [a,b,*,1], Florent Jacquemard [a,b,**]

[a] IRCAM, 1 place Igor-Stravinsky, 75004 Paris, France
[b] Sorbonne Universités, Inria, UPMC Univ Paris 06, IRCAM – CNRS UMR SMTS, Paris, France

ARTICLE INFO

Article history:
Received 18 September 2015
Received in revised form 9 August 2016
Accepted 12 August 2016
Available online xxxx

Keywords:
Model based testing
Interactive music systems
Timed automata

ABSTRACT

The role of an Interactive Music System (IMS) is to accompany musicians during live performances, acting like a real musician. It must react in realtime to audio signals from musicians, according to a timed high-level requirement called mixed score, written in a domain specific language. Such goals imply strong requirements of temporal reliability and robustness to unforeseen errors in input, yet not much addressed by the computer music community.

We present the application of Model-Based Testing techniques and tools to a state-of-the-art IMS, including in particular: offline and on-the-fly approaches for the generation of relevant input data for testing (including timing values), with coverage criteria, the computation of the corresponding expected output, according to the semantics of a given mixed score, the black-box execution of the test data on the System Under Test and the production of a verdict. Our method is based on formal models in a dedicated intermediate representation, compiled directly from mixed scores (high-level requirements), and either passed, to the model-checker Uppaal (after conversion to Timed Automata) in the offline approach, or executed by a virtual machine in the online approach. Our fully automatic framework has been applied to real mixed scores used in concerts and the results obtained have permitted to identify bugs in the target IMS.

1. Introduction

Interactive Music Systems (IMS) [33] are involved in live music performances and aim at acting as an electronic musician playing with other human musicians. We consider such systems that work with a *mixed score*, written in a Domain Specific Language (DSL), which describes the input expected from human musicians, together with the electronic output to be played in response. During a performance, a score-based IMS aligns in real-time the performance of the human musicians to the score, handling possible errors, detects the current tempo, and plays the electronic part. A popular example of this scenario is automatic accompaniment [14].

☆ This work has been partly supported by a DGA-MRIS scholarship and the project Inedit (ANR-12-CORD-009).
* Principal corresponding author.
** Corresponding author.
E-mail addresses: clement.poncelet@ircam.fr (C. Poncelet), florent.jacquemard@inria.fr (F. Jacquemard).
1 DGA Direction Generale de l'Armement (French national defense).

http://dx.doi.org/10.1016/j.scico.2016.08.002
0167-6423/

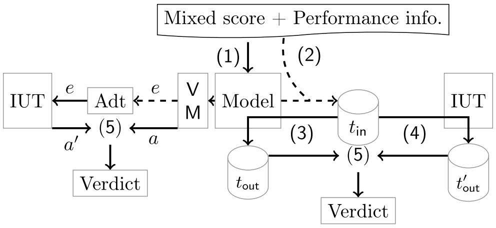

Fig. 1. Highlights of our score-based IMS testing workflows. Left: online; Right: offline.

A score-based IMS is therefore a *reactive* system, interacting with the outside environment (the musicians) under strong timing constraints: the output (generally messages passed to an external audio application such as MAX [32]) must indeed be emitted at *the right moment*, not too late but also not too early. This may be a difficult task since audio calculations often have an important impact on the resource consumptions. In this context, it is important to be able to assess the behavior of an IMS on a given score before its real use in a concert. A traditional approach is to rehearse with musicians, trying to detect potential problems manually, *i.e.* by audition. This tedious method offers no real guaranty since it is not precise, not complete (it covers only one or a few particular musician's performances), and error prone (it relies on a subjective view of the expected behavior instead of a formal specification).

In this paper, we present the application of Model Based Testing (MBT) techniques to a score-based IMS called Antescofo, used frequently in world class concerts in the contemporary repertoire. Roughly, our method proceeds with the steps depicted in Fig. 1. First, (1) a given mixed score is compiled into an Intermediate Representation (IR). This formalism is an executable medium level code modeling the behavior expected from the Implementation Under Test (IUT), the IMS Antescofo, when playing the given mixed score. It has the form of a finite state machine with delays, asynchronous communications and alternations.

Based on this IR, we follow two main approaches for testing: In an *offline* approach (the right side of the figure), the IR is transformed into a Timed Automata (TA) network [3], for generation and simulation purposes. The input test data (2) is generated either by tools from the Uppaal suite [23] for the generation of covering test suites (under restrictions), or either by adding to an *ideal* trace of musician some fuzz of different kinds, or also by translation from a musical performance. Once some timed input traces t_{in} have been generated (representing a musician's performance on the score), the IR is used to compute offline, by simulation (3), the corresponding output traces t_{out}, expected from the system in response to the input. Finally, every input trace t_{in} is sent to the IMS (4) and the real outcome of the IMS, t'_{out} is compared (5) to the expected output t_{out} in order to produce a test verdict.

In an *online* approach (left side of the figure), the IR is executed by a Virtual Machine, and the input and output test data are generated on the fly, using an adapter (Adt). In this approach, the generated input test data is also sent on the fly to the IUT, as the input of an artificial musician, and the real outcome of the IUT is compared online to the expected output (5), event by event. The use of *virtual clocks* permits to execute the tests (in both approaches) in a fast forward fashion, without having to wait for the real duration of the score.

Our case study presents important originalities compared to other MBT applications to realtime systems such as [16,23]. On the one hand, the model supports several time units, including the *physical time* (wall clock), measured in seconds, and the *musical time*, measured in number of beats relatively to a tempo. This situation raises several new problems for the generation of test suites and their execution. On the other hand, the formal model on which our test procedure is based is constructed automatically from a given mixed score, instead of being written manually by an expert – see the discussion on that last point in the related work below. This enables a fully automatic test scenario fitting well in a music authoring workflow where scores in preparation are constantly evolving.

Our main contributions are the design of an appropriate IR, the implementation of a compiler of Antescofo mixed score into IR, and the design of the two above offline and online test procedures based on IR models. Our MBT framework permits the test of the timing behaviors in addition to the output correctness of a system. Applied to Antescofo, it allowed to detect tempo computation errors and synchronization mistakes occurring during non-trivial performances. It is also used to apply regression tests in order to ensure the stability of the system during the development of new versions.

The paper is organized as follows: Section 2 introduces the system under test, Antescofo, and the principles of our test method. The models and their construction from mixed scores are formally defined in Section 3. The implemented model-based testing framework is then described in Section 4. Interesting results, following several options are presented Section 5, and finally perspectives in Section 6.

1.1. Related works

This paper extends an earlier version published in the proceedings of ACM-SAC 2015, track SVT. The main additions compared to the former version are complete semantics of the IR, more details on the compilation of mixed scores into IR and translation of IR into TA, and the presentation of a new online MBT framework, in addition to the offline framework (based on Uppaal) which was presented at ACM-SAC.