

Time-optimal Real-Time Test Case Generation using UPPAAL

Anders Hessel[‡], Kim G. Larsen[†], Brian Nielsen[†], Paul Pettersson[‡], and Arne Skou[†]

[†]Department of Computer Science
Aalborg University
Fredrik Bajersvej 7E
DK-9220 Aalborg, Denmark

[‡]Department of Information Technology
Uppsala University
P.O.Box 337
SE-751 05 Uppsala, Sweden

E-mail: {kgl,bnielsen,ask}@cs.auc.dk, {hessel,paupet}@it.uu.se

Testing is the execution of the system under test in a controlled environment following a prescribed procedure with the goal of measuring one or more quality characteristics of a product, such as functionality or performance. Testing is the primary software validation technique used by industry today. However, despite the importance and the many resources and man-hours invested by industry, testing remains quite ad hoc and error prone.

A promising approach to improving the effectiveness of testing is to base test generation on an abstract formal model of the system under test (SUT) and use a test generation tool to (automatically or user guided) generate and execute test cases. Model based test generation has been under scientific study for some time, and practically applicable test tools are emerging. However, little is still known in the context of hard real-time systems, and few proposals exist that deals explicitly and systematically with testing real-time properties.

A principle problem is that a very large number of test cases (generally infinitely many) can be generated from even the simplest models. The addition of real-time adds another source of explosion, i.e. *when* to stimulate the system and expect response. Thus, an automatically generated test suite easily becomes prohibitively costly to execute.

In this paper we demonstrate how it is possible to generate time-optimal test cases and test suites, i.e. test cases and suites that are guaranteed to take the least possible time to execute. Time optimal test suites are interesting for several reasons. First, reducing the total execution time of a test suite allows more behavior to be tested in the (limited) time allocated to testing; this means a more thorough test. Secondly, it is generally desirable that regression testing can be executed as quickly as possible to improve the turn around time between changes. Thirdly, it is essential for product in-

stance testing that a thorough test can be performed without testing becoming the bottleneck, i.e., the test suite must be applied to all products coming of an assembly line. Finally, in the context of testing of real-time systems, we hypothesize that the fastest test case that drives the SUT to a some state, also has a high likelihood of detecting errors, because this is a stressful situation for the SUT to handle.

We propose a new technique for automatically generating time optimal test cases and test suites for embedded real time systems. We focus on conformance testing i.e. checking by means of execution whether the behavior of some black-box implementation conforms to its specification, and more over doing this with the minimum time. The fact that the SUT is a black-box means that communication with the SUT only takes place via a well defined set of observable actions which implies limited observability and controllability. The required behavior is specified using UPPAAL style timed Automata. The fastest diagnostic trace facility of the UPPAAL model checking tool is used to generate time optimal test sequences.

The test cases can either be generated using manually formulated test purposes or automatically from several kinds of coverage criterion—such as transition or location coverage—of the timed automata model. The coverage approach guarantees that the test suite is derived systematically and that it guarantees a certain level of thoroughness. Even coverage based test suites are guaranteed to be time optimal.

The main contributions of the paper are: definition of a subclass of timed automata from which the diagnostic traces of uppaal can be used as test cases; application of time and cost optimal reachability analysis algorithms to the context of test case generation; a technique to generate time optimal covering test suites; experimental evidence in that our technique has practical merits.