Testing the Temporal Behavior of Real-Time Tasks using Extended Evolutionary Algorithms

Joachim Wegener, Harmen Sthamer
DaimlerChrysler, Research and Technology, Alt-Moabit 96a, 10559 Berlin, Germany
phone: +49 30 39982-232, fax: -107
Joachim.Wegener / Harmen.Sthamer@daimlerchrysler.com

Hartmut Pohlheim
Harmut@pohlheim.com

Keywords: evolutionary testing, testing temporal behavior, evolutionary computation, real-time systems

Abstract

For real-time systems, correct system functionality depends on logical as well as on temporal correctness. Static analysis alone is not sufficient to verify the temporal behavior of real-time systems. Since existing test methods are not specialized for the verification of temporal correctness, we have developed a new testing method, namely evolutionary testing. This paper illustrates results of the first industrial application of the evolutionary test.

1 Introduction

In the course of system design, schedulability analyses are employed to construct systems in a way that these meet the temporal requirements. Techniques of static analysis are used to assess the execution times of tasks as precondition for the schedulability analysis. To date, no commercial tools have been developed for the static analysis of a programs' temporal behavior. Therefore, in practice execution times often have to be estimated through a complex and error-sensitive manual inspection of the code [4]. Even if professional tools for the static computation of execution times were to be available in the future, errors can not be ruled out of time estimations, e.g. caused by:

- errors in the implementation of static analysis tools,
- incomplete information on the dynamic behavior of the target processor, or
- false information given by the developers on the program's dynamic features, e.g. maximum number of iterations of program loops.

Therefore, as long as no formal verification methods exist, systematic testing is an inevitable part of the verification and validation process for real-time systems. We have developed a new approach for testing temporal behavior, namely evolutionary testing. In various experiments evolutionary testing has been compared to static analysis [2], random testing [6], and systematic testing [7] and achieved good results.

Evolutionary operators used for evolutionary tests have been constantly improved. The use of extended evolutionary algorithms overcomes the problems reported in using genetic algorithms. This establishes the foundation of the first industrial application of the evolutionary test for the dynamic test of a motor control system.

2 Testing Temporal Behavior

The major objective of testing is error detection. The temporal behavior of real-time systems is defective when input situations exist in such a manner that their computation violates the specified timing constraints. In most cases the task of the tester therefore is to find those input situations with the shortest or longest execution times to check whether they produce a temporal error. Within evolutionary testing the search for the shortest and longest execution times is regarded as an optimization problem to which evolutionary algorithms are applied. With the exception of very simple real-time systems, temporal behavior always forms a very complex multi-dimensional search space with many plateaus and discontinuities. Due to the complexity of the temporal behavior, it is not astonishing that some researchers in control theory have drawn the conclusion that computer systems are inherently probabilistic in terms of their timing behavior [5]. Correspondingly difficult is the testing of temporal behavior with conventional black-box and white-box test methods. The evolutionary test therefore grounds on a stochastic procedure: evolutionary algorithms.

3 Evolutionary Testing

Evolutionary algorithms represent a class of adaptive search techniques and procedures based on the processes of natural genetics and Darwin's theory of evolution. They are characterized by an iterative procedure and work in parallel with a number of potential solutions, the population of individuals. In every individual, permissible solution values for the variables of the optimization problem are coded. The evolutionary search and optimization process is based on the three fundamental principles selection, recombination and mutation.

When using evolutionary algorithms for determining the shortest and longest execution times of test objects, each individual of the population represents a test datum with which the system under test is executed. For every test datum, the execution time is measured to determine the fitness value of each individual.

The evolutionary process repeats itself until a given stopping condition is reached, e.g. a certain number of generations, or when an execution time is found which is outside the specified timing bounds of the system under test. In this case, a temporal error is detected. If, however, all the times found meet the timing constraints, confidence in the temporal correctness of the system is substantiated.

Our work applies the extended evolutionary testing [6] which allows the combination of multiple strategies, e.g. global and local searches and the automatic distribution of resources in accordance with the success of the strategies. For a detailed discussion of evolutionary algorithms, see [1] or [3]. Figure 1 shows the structure of the evolutionary process.

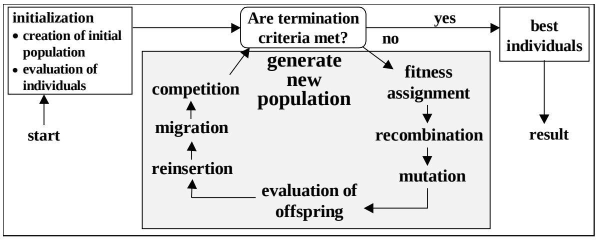

Figure 1: Structure of the extended evolutionary algorithm

4 Testing the Motor Control System

The application field is a new motor control system for six- and eight-cylinder blocks that is currently under development. It contains several tasks that have to fulfil timing constraints. The execution times for these tasks are determined by the use of hardware timers in the target environment. They have a resolution of 400 ns. A hardware timer is questioned directly before and after the execution of the tested task to determine its execution time.

Evolutionary testing (ET) is used to verify the results from the developers' tests (DT) which are based on the functional specification of the system as well as on the internal structures of the tasks. The longest execution times determined by the evolutionary testing are compared to the maximum execution times found by the developers' tests. The tests are performed on the target processor later used in the vehicles. The test results are shown in Table 1.

Task name	Max. execution time in µs		Lines of code	Input parameter
	ET	DT		
Task zr2	69,6 µs	67,2 µs	41	18
Task t1	120,8 µs	108,4 µs	119	18
Task mc1	112,0 µs	108,4 µs	98	17
Task mr1	68,8 µs	64,0 µs	81	32
Task k1	59,6 µs	57,6 µs	39	14
Task zk1	58,4 µs	54,0 µs	56	9

Table 1: Maximum execution times of motor control tasks

It can be clearly seen that evolutionary testing has found longer execution times for all the given tasks. This is surprising for evolutionary testing treats the software as black boxes, whereas the developers are familiar with the function and structure of their system. An explanation might be the use of system calls, compilation and optimization, as well as dependencies on other system components, of which the temporal effects can only be rated with difficulty by the developers. With regard to the motor control system execution times determined by evolutionary testing do not exceed the temporal constraints of a task in any of the cases. The irregularities of the test results therefore have not been disquieting. The intensive testing of the temporal behavior with systematic and evolutionary tests has strengthened the developers' confidence in a correct temporal behavior of the system.

5 Conclusion and Future Work

Temporal correctness is crucial to the flawless functioning of real-time systems. Our work investigated the use of extended evolutionary algorithms to validate the temporal correctness of a motor control system.

For all the tasks of the motor control system, evolutionary testing achieved better results in comparison with the developers' tests. This illustrates how difficult it is to thoroughly test the temporal behavior of systems by using systematic testing. However, since no search strategy can guarantee that extreme execution times will be found, the use of evolutionary testing alone is not sufficient for a thorough and comprehensive examination of temporal behavior. A test strategy for real-time systems should at least include systematic testing and evolutionary testing.

In future, we intend to examine this combination with static analysis more closely [2]. By combining both approaches, the area in which one finds the extreme execution time of the system can be closely defined. This means, developers of real-time systems would gain an effective tool, to rate exactly the minimum and maximum execution times for their systems.

References

[1] *Mitchell, M.*: An Introduction to Genetic Algorithms. Cambridge, Massachusetts: MIT Press, 1996.

[2] *Mueller, F. and Wegener, J.*: A Comparison of Static Analysis and Evolutionary Testing for the Verification of Timing Constraints. Proceedings of the IEEE Real-Time Technology and Applications Symposium RTAS '98, pp. 144-154, 1998.

[3] *Pohlheim, H.*: Entwicklung und systemtechnische Anwendung Evolutionärer Algorithmen. Aachen, Germany: Shaker Verlag, 1998. (Development and Engineering Application of Evolutionary Algorithms. Ph.D. thesis, in german)

[4] *Puschner, P., and Vrchoticky, A.*: Problems in Static Worst-Case Execution Time Analysis. Proceedings of the 9th ITG/GI-Conference on Measurement, Modeling and Evaluation of Computational and Communication Systems, pp. 18-25, 1997.

[5] *Törngren, M.*: Fundamentals of Implementing Real-Time Control Applications in Distributed Computer Systems. In Wikander, J., and Svensson, B. (Eds): Real-Time Systems in Mechatronic Applications, Kluwer Academic Publishers, Boston, USA, 1998.

[6] *Wegener, J., and Grochtmann, M.*: Verifying Timing Constraints of Real-Time Systems by Means of Evolutionary Testing. Real-Time Systems, 15, pp. 275-298, 1998.

[7] *Wegener J., Sthamer H., and Pohlheim H.:* Testing the Temporal Behavior of Real-Time Tasks using Extended Evolutionary Algorithms. EUROSTAR'99, Barcelona, Spain 1999.