ARTES project status report August ‘99
TATOO

Test And Testability Of Distributed Real-time Systems

Hans Hansson and Henrik Thane

Department of Computer Engineering
Mälardalens University College
Box 883, 721 23 Västeras
han@idt.mdh.se

1 Summary

This is a report of the achievements and plans of the TATOO project which started October 1st, 1998, and which has been granted support from ARTES for one graduate for one year. An application for extending the support for a second year will be submitted to ARTES in August 1999.

The goal of the research is to develop methods, metrics, design rules, and tools for testing of distributed real-time systems (DRTS). Theoretical and practical results with regard to testing and testability of DRTS are next to nonexistent. Research is therefore of significance to both academia and Industry. During the first phase of the project we have developed: Methods for achieving deterministic testing of DRTS, testability metrics, and tools that implements the results. We have specifically addressed task sets with recurring release patterns, executing in a distributed system, where the scheduling on each node is handled by a priority driven preemptive scheduler. This includes statically scheduled systems that are subject to preemption [15] and interrupts [8], as well as strictly periodic fixed priority systems [1][5].

2 Problem Statement

Achieving deterministic testing of sequential programs is easy because we need only control the sequence of inputs and the start conditions, in order to guarantee reproducibility [6]. That is, given the same initial state and inputs, the sequential program will deterministically produce the same output on repeated executions, even in the presence of systematic faults [7]. Reproducibility is essential when performing regression testing or cyclic debugging [9], where the same test cases are run repeatedly with the intent to validate that either an error correction had the desired effect, or simply to make it possible to find the error when a failure has been observed [3]. However, trying to directly apply test techniques for sequential programs on distributed real-time systems is bound to lead to non-determinism and non-reproducibility, because control is only forced on the inputs, disregarding the significance of order and timing of the executing and communicating tasks. Any intrusive observation of a DRTS will in addition incur a temporal probe-effect [2][4] that subsequently will affect the system’s temporal and functional behavior.

In order to achieve systematic testing of DRTS there are three major problems that need to be addressed:

(1) Reproducing the inputs with respect to contents, order, and timing

(2) Deterministically observing or reproducing the order and timing of the execution of the parallel programs as well as their communication with each other and the environment

(3) Eliminating the probe-effect.

In the TATOO project we target problems (2) and (3). For a description of related research regarding these problems see Thane and Hansson [11].

3 Main Ideas

The main ideas of TATOO are to:

1. Facilitate deterministic testing of DRTS by transforming the non-deterministic DRTS testing problem into a set of deterministic sequential programs testing problems. This is can be achieved by deriving all the possible execution orderings of the distributed real-time system and regarding each of them as a sequential program. We intend to specifically addressed task sets with recurring release patterns, executing in a distributed system,

where the scheduling on each node is handled by a priority driven preemptive scheduler. This includes statically scheduled systems that are subject to preemption [15] and interrupts [8], as well as strictly periodic fixed priority systems [1][5]. Together with an accompanying testing strategy this approach could allow test methods for sequential programs to be used, since each identified ordering can be regarded as a sequential program. We intend to take into account the effects of interrupts, preemption, clock synchronization, and varying start and execution times of the involved tasks.

2. Develop testability metrics that gives criteria on when to stop testing, and what to test. This is important since any criteria less than exhaustive testing must be justifiable. A metric is also useful for analysis, and comparison between different design solutions and architectures, so that sound design rules can emerge.

3. Eliminate the probe effect through the allocation of sufficient resources and then letting the probes remain in the target system. This includes allocating resources for the probes' execution time, memory, communication bus bandwidth and accounting for the probes when designing and scheduling. In order to guarantee consistent observations of the global state in the DRTS we assume that the system is globally scheduled. Which means that the release and execution times can be related to a global synchronized time base with a known precision.

Since research into this field, in the real-times systems community, has been meager we intend to adapt, where possible, results from testing of sequential programs and testing of concurrent programs (which is quite meager too), and fill in the blanks where there are no relations what so ever.

4 Scientific merits

The result will be a set of methods, and tools, for testing distributed real-times systems, addressing what current test methods for sequential programs cannot test. Corresponding to the set of methods there will be a set of testability metrics, and tools, for finding out how many test-cases are necessary in order to find all errors in the code with a certain confidence. In close relation to the testability metrics there will be methods for identification of the actual test-cases that must be executed in order to satisfy the coverage criteria defined by the test methods and testability metrics.

The prospects of finding such methods and metrics are quite good when the semantical restrictions on distributed real-time systems, based on static scheduling or fixed priority scheduling, are significantly greater than those for concurrent systems where no notion of real-time exists.

There is very little done (next to nothing) in this field, so if the project is fruitful the impact will be considerable both to Academia (potentially opening up a new field of research) and to the Industry where there is a dire need for methods and tools.

5 Project Plan and Results

The project is performed by Henrik Thane (a Ph.D. student that has previously studied the problem area in his Licentiate thesis [12]), and supervised by Hans Hansson. The project is expected to span an 18-24 months period (support from ARTES for the first half of the project has been granted).

Month 1-10: Project commenced in October 1998 and this period has passed (Oct. – July). Related work has been studied and collected. Two papers have been produced [10][11], and two technical reports are in the works [13][14]. The paper [11] has been accepted for publication at the IEEE Real-Time Systems Symposium conference in December 1999. The other paper[10] has been submitted to the RTCSA conference in December 1999. A tool that implements the results in the papers has been produced. A master's level exam project with two students designing and implementing a real-time kernel with monitoring mechanisms has commenced and is expected to complete before September 1999.

Month 11-17: (Aug.- Feb. 1999) Thesis work begins. Extend theory in [10][11] with critical regions. As soon as the experimental real-time kernel is available begin experimenting with the monitoring mechanisms described in [14] as well as trying to empirically validate the theoretical claims in [10][11][13], including the application of results to real industrial applications obtained from our industrial partners. 1-2 conference articles will be submitted.

Month 18-22: Writing the PhD thesis.

Month 23-24: Prepare and wait for dissertation day; write a journal article.

6 Related Research

See produced paper [11] for related research.

7 Industrial relevance and exploitation expectations

Our proposed research aims at solving real-time problems stemming from industrial demand. It is carried out in co-operation with the industrial partners *Volvo Construction Equipment Components AB*, *Datex-Ohmeda AB*, and *Enator Teknik Mälardalen AB*. The research will provide us with insights in concrete, application specific, real-time engineering practices and broaden the scientific foundation for industry.

Christer Eriksson and Kristian Sandström at Mälardalen University have developed a scheduling tool that takes real-world constraints into consideration. This tool is in use at Volvo Construction Equipment Components AB. The experiences gained from that project have been very positive. However, integrating this work with testing methods and testability analysis tools could be of benefit for the designers by providing criteria on when to stop testing, what to test, how to test and how to design for high testability.

In TATOO we expect to learn how to design and test real-world real-time systems by analyzing the actual problems, their requirements and used industrial practice, and by providing solutions and research to meet the encountered industrial demands.

8 References

[1] Audsley N. C., Burns A., Richardson M.F., and Wellings A.J. Hard Real-Time Scheduling: The Deadline Monotonic Approach. IFAC/IFIP Worshop, Atlanta, Georgia, pp. 127-132, May, 1991

[2] Gait J. A Probe Effect in Concurrent Programs. Software – Practice and Experience, 16(3):225-233, Mars, 1986.

[3] Laprie J.C. Dependability: Basic Concepts and Associated Terminology. Dependable Computing and Fault-Tolerant Systems, vol. 5, Springer Verlag, 1992.

[4] LeDoux C.H., and Parker D.S.. Saving Traces for Ada Debugging. In the proceedings of Ada int. conf. ACM, Cambridge University press, pp. 97-108, 1985.

[5] Lui C. L. and Layland J. W.. Scheduling Algorithms for multiprogramming in a hard real-time environment. Journal of the ACM 20(1), 1973.

[6] McDowell C.E. and Hembold D.P. Debugging concurrent programs. ACM Computing Surveys, 21(4), pp. 593-622, December 1989.

[7] Rushby J. Formal methods and their Role in the Certification of Critical Systems. 12th Annual CSR Workshop, Bruges 12-15 September 1995. Proceedings, pp. 2-42. Springer. ISBN 3-540-76034-2.

[8] Sandström K., Eriksson C., and Fohler G. Handling Interrupts with Static Scheduling in an Automotive Vehicle Control System. In proceedings of the 5th Int. Conference on Real-Time Computing Systems and Applications (RTCSA'98). October 1998, Japan.

[9] Schütz W. Fundamental Issues in Testing Distributed Real-Time Systems. Real-Time Systems, vol. 7(2), pp. 129-157, 1994.

[10] Thane H, Hansson H. Towards Deterministic Testing of Distributed Real-Time Systems. Real-Time Computing Systems and Applications conference (RTCSA'99), Hong Kong China, December, 1999. (Submitted)

[11] Thane H, Hansson H. Towards Systematic Testing of Distributed Real-Time Systems. To appear in Real-Time Systems Symposium, Phoenix U.S.A, December, 1999.

[12] Thane H. Safety and Reliability of Software in Embedded Control Systems, Technical Licentiate thesis, Royal Institute of Technology, Stockholm, Oct. 1997.

[13] Thane H. Towards exact minimum and maximum response time analysis for preemptive strict periodic fixed priority scheduled distributed real-time systems using execution ordering graphs. Technical report, Mälardalen Real-Time Research Centre, Dept. Computer Engineering, Mälardalen University, 1999.

[14] Thane H. Towards predictable instrumentation of distributed real-time systems. Technical report, Mälardalen Real-Time Research Centre, Dept. Computer Engineering, Mälardalen University, 1999.

[15] Xu J. and Parnas D. Scheduling processes with release times, deadlines, precedence, and exclusion, relations. IEEE Trans. on Software Eng. vol. 16, pp. 360-369, 1990.