ARTES Project Extension Application
TATOO

Test And Testability Of Distributed Real-time Systems

Henrik Thane

Mälardalen Real-Time Research Centre
www.mrtc.mdh.se/projects/tatoo
Department of Computer Engineering
Mälardalen University
hte@mdh.se

Introduction

This is a request for an extension of the funding of the TATOO project, with 2 years of funding *for one **new** graduate student*. The project has previously been granted funding for two years, and started October 1st, 1998, with Henrik Thane as graduate student, who has now completed his Ph.D. thesis.

The project deals with fundamentals for deterministic testing of distributed real-time systems (DRTS), as well as methods for deterministic debugging and monitoring of DRTS. Theoretical and practical results with regard to testing and debugging of DRTS were next to nonexistent before the start of the project, and research was therefore deemed significant to both academia and Industry.

During the first 19 months of the project (since October 1st 1998) we have developed: Methods for achieving deterministic testing of DRTS [1][6][7][10][5], methods for deterministic debugging [2] and methods for deterministic monitoring of DRTS [8]. We have also produced tools that implement the results and developed supporting infrastructure [4]. The results have been reported in a number of publications (see list of publications) and one Ph.D. thesis [2].

As the initial results of the project have shown the potential of producing novel results in this unexplored area, we therefore propose an extension of the project in order to further penetrate the area and refine the initial results.

Industrial exploitations

The initial phase of the project has bared fruit and the prospects of commercializing the results are promising. A spin-off company is in the process of being started. There is a dire need in the industry for testing, debugging and monitoring tools for DRTS. Our industrial partners give us support and we will now proceed and evaluate the initial results in cooperation with our industrial partners: Datex-Ohmeda, Volvo Construction Equipment, and Enator Teknik Mälardalen.

Project Plan

The project team will consist of Henrik Thane (Project Leader), Hans Hansson and a new Ph.D. candidate. The continuation of the project is expected to span a 24-30 months period beginning August 1st 2000.

Expected milestones:

- Implementation of prototype tools and evaluation of these in cooperation with one of our industrial partners in industrial strength case-studies. We will specifically address the results in the papers on:
  - Deterministic *debugging* [2] and *monitoring* [8] of DRTS using the specifically developed real-time kernel Asterix [4].
  - Deterministic *testing* [1][6][7][10] and *monitoring* [8] of DRTS using the specifically developed real-time kernel Asterix [4].
- Extension and refinement of the theory in papers [1][6][7][10] with respect to attaining tighter coverage estimates for testing of DRTS as well as minimizing the pessimism in the coverage requirements when allowing interrupts in DRTS as suggested in the Ph.D thesis [2].

Preliminary budget

We initially ask for funding of one new Ph.D. student for a two-year period, i.e., a total of 1.2 MSEK.

More information

For more information about the project see appended documents: (1) ARTES project status report August ’99, (2) the previous ARTES project proposal, (3) the original suggestions by the ARTES board and our comments to these, the Ph.D thesis (4), or the publications listed in this proposal, (5) Support Letter, (6) Henrik Thane’s CV.

Publications

[1] Thane H and Hansson H. *Testing Distributed Real-Time Systems.* Submitted for journal publication.

[2] Thane H. *Monitoring, Testing and Debugging of Distributed Real-Time Systems*. Ph.D. Thesis. May 2000.

[3] Thane H. and Hansson H. *Using Deterministic Replay for Debugging of Distributed Real-Time Systems*. In proceedings of the 12th Euromicro Conference on Real-Time Systems (ECRTS’00), Stockholm, June 2000.

[4] Thane H. *Asterix the T-REX among real-time kernels. Timely, reliable, efficient and extraordinary.* Technical report. In preparation. Mälardalen Real-Time Research Centre, Mälardalen University, May 2000.

[5] Thane H, and Wall A. *Formal and Probabilistic Arguments for Reuse and testing of Components in Safety-Critical Real-Time Systems.* Technical report. Mälardalen Real-Time Research Centre, Mälardalen University, March 2000.

[6] Thane H. and Hansson H. *Handling Interrupts in Testing of Distributed Real-Time Systems*. In proc. Real-Time Computing Systems and Applications conference (RTCSA’99), Hong Kong, December 1999.

[7] Thane H. and Hansson H. *Towards Systematic Testing of Distributed Real-Time Systems*. In proceedings of the 20th IEEE Real-Time Systems Symposium (RTSS’99), Phoenix, Arizona, December 1999.

[8] Thane H. *Design for Deterministic Monitoring of Distributed Real-Time Systems.* Technical report, Mälardalen Real-Time Research Centre, Mälardalen University, November 1999.

[9] Norström C., Sandström K., Mäki-Turja J., Hansson H., and Thane H. *Robusta Realtidssytem*. Book. MRTC Press, September, 1999.

[10] Thane H. and Hansson H. *Towards Deterministic Testing of Distributed Real-Time Systems.* In Swedish National Real-Time Conference SNART'99, August 1999.

Appendix (1)

ARTES project status report August ’99

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 systems 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 masters 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.

Appendix (2)

The original project proposal

Proposal to ARTES
TATOO

Test And Testability Of Distributed Real-time Systems

Hans Hansson

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

1 Summary

The goal of the proposed research is to develop metrics for testability, test methods, tools and design rules for high testability in distributed real-time systems (DRTS). Theoretical and practical results with regard to testing and testability of distributed real-time systems are next to nonexistent. Research is therefore of significance to both academia and Industry. We propose here a project for development of methods and metrics for testing and measurement of testability of software in distributed real-time systems.

2 Problem Statement

There is a lack of methods and techniques for testing and measuring testability of software in distributed real-time systems. Bluntly put, it is a black hole. Current theory and practice for testing sequential programs are simply not good enough, due to complex issues like: probe effects and race conditions – intrinsic to all concurrent systems, and lack of synchronized global clocks, that lead to non-reproducible behavior and higher complexity. Still, for safety-critical computer control systems (which all are real-time systems) we must design and verify the reliability and safety of the software to very high standards – but have no means to do so without great expense (and still no guarantees.)

A Stoppage Criteria

The question of when to stop testing has never been adequately answered. One answer is ”test until you cannot test any further”, either due to lack of time or money. Others argue that we must satisfy 100% of some structural coverage criterion. For example, attempt 100% branch coverage which is terminated when it is satisfied. Only exhaustive testing is truly thorough, so any stoppage criteria that is applied before exhaustive testing is satisfied must be justifiable – that is for reasons better than ”we ran out of time and money” [Voas1995].

With the help of a testability metric and a corresponding testing methodology we can give an answer to when to stop testing. That is, the testability metric should give an indication on how many test-cases are necessary to find all the errors of types limited by the ability of the testing methodology.

If we, with the help of a testability metric, finds out that the number of test-cases needed are simply too many to execute, we can apply some other form of verification, like reviews or formal verification. But, we can also deem if it is more suitable to redesign the system for higher testability.

Design for High Testability

Software does not wear out over time. It is therefore reasonable to assume that as long as faults are uncovered – reliability increases for each fault that is eliminated [Littlewood1973]. A problem with this method is that it would take years to remove a sufficient amount of errors to achieve a critical standard of reliability. For safety-critical systems where the failure rate is required to be lower than 10^{-9} failures per hour this is an intractable task since testing has to be performed for at least 115 000 years [Fenton1995].

It is therefore imperative to design for high testability in order to decrease the number of test cases needed for verifying that the software is of ultra-reliable or safety-critical standards (10^{-9}).

Iterative Design Process

What makes matters even worse is the fact that more than half of the errors in a system are due to ambiguous or incomplete requirement specifications [Leveson1995, Lutz1992, DeMarco1978, Ellis1995]. The intention of testing is to verify that a specific input will yield a specific output, defined by the *specification*. Thus the confidence gained by testing software is limited. That is, we can by testing not reveal errors in the requirements.

Requirements are the main obstacle when designing and verifying software. This has led to an iterative development process where the requirements are refined for each iteration and prototypes produced – which then are validated against the customer. Using this scheme, the portion of failures related to inadequate requirements can be decreased.

However, for each iteration and prototype we must re-test the system. High testability is then of utmost importance in order to decrease the testing effort for every iteration. That is, it is desirable to design for high testability.

Currently no test methods and testability metrics for distributed real-time systems

With all these arguments at hand, we are back to what we stated in the first paragraph in this section: there are no testability metrics and testing methods that addresses the specifics of distributed real-time systems. So before we have them we cannot design for high testability and not test safety-critical systems software to the standards demanded by the customers, the public and the legislators.

3 Main Ideas

We propose to:

1. Develop a testability metric that gives criteria on when to stop testing, and what to test. This is important since we must know when we have tested an application enough, or rather when we have not. A metric is also imperative for enabling analysis, and comparisons between different design solutions and architectures, so that sound design rules can emerge.

   The metric will be based on a model that takes into account the effects, that different execution and communication strategies for DRTS, have on testability. For example, the consequences of the use of time-triggered or event-triggered architectures, or what the effects are if preemption, interrupts and jitter are allowed, or what happens when you mix different architectures, like time triggered nodes in a system which is interconnected by an even-triggered network – leading to a non homogeneous DRTS.

2. Develop test methods, for systems run by static (off-line) or fix priority (on-line) scheduling principles, that addresses the specific problems of testing concurrent and time dependent systems.

   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 Expected Results and Impact

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, that state how many test-cases are necessary – that is, giving stoppage criteria, stating the minimum number of test-cases needed for finding all the errors in the code, with a certain confidence. In close relation to the testability metrics there will be methods that identify 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 (foremost) or fix-priority scheduling (to a lesser extent), are significantly greater than those for concurrent systems where no notion of real-time exists.

To the authors' big surprise 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.

6 Project Plan

This project will be conducted by a Ph.D. student with 25% of department duties and supervised by Hans Hansson and Christer Eriksson. This means that the project will span over a 30-33 months period.

Month 1-6: Start of project. Formalize why it is difficult to test concurrent systems in general, and distributed real-time systems especially. Begin state-of-the-art paper addressing testing and testability-metrics for DRTS, by collecting related work. Make preliminary model for testability metric of statically scheduled DRTS, and define preliminary testing methods as well as methods for test-case generation. Start building prototypes.

Month 7-10: Analysis of prototypes. Apply prototypes to Industrial software, collect data, and document case-study. Refine metrics, test methods, and test-case generation. Define suitable design rules for high testability.

Month 11-17: Make preliminary model for testability metric of fix priority scheduled DRTS, and define preliminary testing methods as well as methods for test-case generation. Start building prototypes for fixed priority scheduling.

Month 18-20: Analysis prototypes for fix priority scheduled DRTS. Apply prototypes to Industrial software, collect data, and document case-study. Refine metrics, test methods, and test-case generation. Define suitable design rules for high testability.

Month 21-30: Write Ph.D. thesis. Including state of the art, state of the practice, models and methods, case-studies with data, analysis of how different architectures affect testability, define suitable design rules for high testability and define future work.

This schedule will be extended with a 1-3 months research stay at a highly ranked international institution. During the 30 months the Ph.D. student will also gather about 20 points of graduate courses in relevant areas.

Henrik Thane has studied the problem area, which was part of his Licentiate thesis work, and has expressed interest to work on the project for his Ph.D. thesis. His CV is included in the appendix.

7 Preliminary Budget

We estimate the total project budget to amount to 1.2 MSEK. This sum is comprised of funding for

- one Ph.D. student salary for 2.5 years, considering departmental duties of 0.5 years
- supervision cost
- research stay for Ph.D. student at international institution (1-3 months)
- One big screen PC, for design and evaluation of test and metrics tools
- conference travels for presentation of results
- a 20% department overhead is added to the above listed items

8 Related Research

The knowledge and research into testing of sequential programs have been considerable. Theory and practice for testing sequential programs is however not applicable to concurrent programs [Helmbold1996] and for distributed real-time systems [Schütz1992], due to race conditions, probe effects and lack of synchronized clocks. The work by Schütz is the sole one that describes the intrinsic problems of testing DRTS.

Promising work on testability for sequential programs, that might to some extent be possible to utilize in testability metrics for DRTS, has been done by Voas [Voas1992, Voas1993], by providing statistical methods for evaluating the testability of sequential programs, and providing a testing stoppage criteria.

Work related to testing of concurrent programs (ADA) that enable reproducible behaviors of the programs has been done by Tai [Tai1991]. This is done by forcefully synchronizing the concurrent processes in the system according to specific scenarios. Tai's work does however, affect the timing in the system, and is as such not directly applicable to real-time systems. The method has however potential to, with modifications, be applicable to fix priority scheduled systems.

Work related to monitoring of distributed real-time systems has been done by [Dodd1992, Tsai1996], which mostly covers how to record events in distributed real-time systems for later off-line analysis, and playback.

9 Relation to Profile and Industry

Our proposed research aims at solving real-time problems stemming from industrial demand. It will be carried out in co-operation with the industrial partners. The research will provide us with insights in concrete, application specific real-time engineering practices and broaden the scientific foundation for industry.

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 demands.

The testability metrics acknowledge the demands for objective measurements of how, for example, architectures for DRTS affect the testability. This can in the end be used for devising rules on how to design and specify DRTS. Methods for testing DRTS acknowledge the demand for reliable and safe systems.

10 Context

10.1 The research group

The proposed research will be carried out at the Computer Engineering Department (IDt) at Mälardalen University, Västeras, in co-operation with Volvo construction equipment, Datex-Engström, and Enator. It is intended to fit with other successful research efforts at IDt for Volvo, in particular the already developed scheduling tool by Christer Eriksson and Kristian Sandström.

Real-time Systems research was established at MdH around 1990 by three persons as a glue between academia and industry. It has since then been the most active and fastest growing research area, as well as a strong educational profile, at MdH. The real-time research at MdH is supported by a number of regional and national authorities, foundations and companies. As a result, real-time systems has been identified (by the university board) as the research area of highest priority at MdH. It has grown to approximately 30 people providing 400 full time student equivalents, 17 people are involved in research.

10.2 Complementary activities

4 student exam projects are currently being defined to aid in developing technology and experimental prototypes.

10.3 Research and industrial co-operation

During the last year 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 testability metrics and testing methods could be of benefit for the designer by providing criteria on when to stop testing, what to test, how to test and how to design for high testability.

Contact person at Volvo Construction Equipment Components AB:

Nils-Erik Bånkestad
Dept. TUE
Volvo Construction Equipment Components AB
631 85 Eskilstuna
vcecmp.neb@memo.volvo.se

Contact person at Datex-Engström AB:

Joakim Fröberg
Datex-Engström
Box 20109
161 02 Bromma
Joakim.Froberg@se.datex-engstrom.com

Contact person at Enator:

Mikael Gustafsson
Enator
Västerås
vcecmp.mikaelg@memo.volvo.se

11 References

[DeMarco1978] T DeMarco. Structured Analysis and System Specification. Yourdon Press 1978. ISBN 0-917072-07-3

[Dodd1992] P. S. Dodd, et. al. *Monitoring and debugging distributed real-time programs*. Software-practice and experience. Vol. 22(1), pp. 863-877, October 1997.

[Ellis1995] A. Ellis. Achieving Safety in Complex Control Systems. Proceedings of the Safety-Critical Systems Symposium. Pp. 2-14. Brighton, England, 1995. Springer-Verlag. ISBN 3-540-19922-5

[Fenton1995] N E Fenton. The Role of Measurement in Software Safety Assessment. 12th Annual CSR Workshop, Bruges 12-15 September 1995. Proceedings, pp. 217-248. Springer. ISBN 3-540-76034-2.

[Helmbold1996] D.P. Helmbold, C.E. McDowell. *A Taxonomy of race conditions*. Journal of parallel and distributed computing. Vol. 33(2), pp. 159-164, March 1996.

[Leveson1995] N. G. Leveson. Safeware System, Safety and Computers. Addison Wesley 1995. ISBN 0-201-11972-2.

[Littlewood1973] B Littlewood et. al. A Bayesian Reliability Growth Model For Computer Software. Journal of the Royal Statistical Society, Series C, No. 22, p 332-346, 1973

[Lutz1992] R. R. Lutz. Analyzing software requirements errors in safety-critical, embedded systems. In software requirements conference, IEEE, January 1992.

[Schütz1992] W. Schütz. The testability of Distributed Real-Time Systems. Ph.D. dissertation. Technische Universität Wien, 1992.

[Tai1991] K.C Tai, et. al. *Debugging concurrent Ada programs by deterministic execution*. IEEE transactions on software engineering. Vol. 17(1), pp. 45-63, January 1991.

[Tsai1996] J.P. Tsai, et. al. *A system for visualizing and debugging distributed real-time systems with monitoring support*. Journal of Software engineering and Knowledge engineering. Vol. 6(3), pp. 355-400, 1996.

[Voas1992] J.M. Voas *PIE: A Dynamic Failure-Based Technique.*. IEEE Transactions on Software Engineering, Vol.18(8), pp.1261-1269, August 1992.

[Voas1993] J.M. Voas, et.al. Designing programs that are less likely to hide faults. Journal of systems and software. Vol. 20, pp. 93-100, 1993.

[Voas1995] J M Voas, et.al. Software Assessment: Reliability, Safety, testability. Wiley Interscience, 1995. ISBN 0-471-01009-x.

12 Appendices

Appendix A: C.V. of Hans Hansson

Appendix B: C.V. of Henrik Thane

Appendix C: Supporting letters

Appendix (3)

The original suggestions by the ARTES board and our comments to these

Suggestions by the ARTES board 1998 and follow up 1999

Suggestions by the ARTES board

Date: June 12th 1998

"ARTES styrelse beslutade om stöd för en ettårig förstudie (max 500kSEK) med anledning av projektförslaget "TATOO - Test And Testability Of Distributed Real-Time Systems". Under denna förstudie ska (bl.a.) kontakter med datorteknik på Chalmers (.) samt IDA vid LiTH (Mary Helander mfl.) tas. Littlewoods arbeten och existerande kommersiell programvara ska också beaktas. Kontakta Bertil Emmertz på ABB ISY (numera Industrial Productions; tror jag) för information om det senare."

Contact or consider the work by:

(1) Martin Hiller, et. al. At the department of Computer engineering, Chalmers

(2) Mary Hellander, et al. At Lindköping University

(3) Bertil Emmertz, ABB Industrial Productions. Consider the work by Littlewood as well as existing commercial software for testing

Follow up on the suggestions

Date: August 18th 1999

The work by (1), and (2), have been considered, although any concrete relevance to the TATOO project could however not be found. Contact with Bertil Emmertz has occurred. The work by Littlewood has been studied earlier by Henrik Thane for his Licentiate thesis. With respect to existing commercial software, the theoretical results of the first phase (year one) of the TATOO project has shown that existing methods and techniques for testing of sequential software can be used after serialization of the distributed real-time system into a set of sequential programs.

Appendix (4)

Henrik Thane's Ph.D Thesis

Appendix (5)

Support Letter

00-05-22

Letter of support

Datex-Ohmeda AB would like to continue supporting the TATOO (Test And Testability Of distributed real-time systems) research project.
A significant part of the development of our products goes into verification, validation and debugging of non-sequential programs. It would be of great use to us, to have monitoring mechanisms, tools for deterministic replay and testing available for our development teams. We would gladly provide industrial experience and real-world applications as input to this project, and want to make use of the results.

Joakim Fröberg

08- 555 221 00

Appendix (6)

Henrik Thane’s CV

CURRICULUM VITAE

Henrik Thane

Friday, May 19, 2000

Name:	*Henrik Carl Gustaf Thane*
Date of birth:	*February 19, 1970*
Place of birth:	*Västerås, Sweden*
Citizenship:	*Swedish*
Telephone:	*+46 21 103157*
Mobile:	*+46 70 7768997*
Fax:	*+46 21 103110*
E-mail:	*henrik.thane@mdh.se*
www:	*http://www.idt.mdh.se/personal/hte/*

Education

May 2000:	**Technical Doctoral** degree in Mechatronics from the Royal Institute of Technology, Stockholm. *Thesis title*: ”Monitoring, Testing, and Debugging of Distributed Real-Time Systems". Dissertation day, May 26.
October 1997:	**Technical Licentiate** in Mechatronics from the Royal Institute of Technology, Stockholm. *Thesis title*: ”Safety and Reliability of Software in Embedded Control Systems ".
November 1995:	**Master of Science** in Computer Science from Uppsala University, Uppsala. *Thesis title*: ”Distributed Realt-Time Clock Synchronization on the CAN Bus".
November 1993:	**Bachelor of Science** in Computer Engineering from Mälardalen University College, Västerås.
June 1990:	High school diploma (4årig teknisk linje), Wenströmska gymnasiet, Västerås
June 1989:	High school diploma, Lakeland High school, New York, USA.

Employment record

1997-08 –	**Ph.D. Student and Lecturer** at the Department of Computer Engineering, Mälardalen University College, Västerås.
1995-04 – 1997-07	**Ph.D. Student** at the Mechatronics Laboratory, the Royal Institute of Technology, Stockholm.
1991-04 – 1994-10	**Programmer and Consultant** of real-time systems software at NST Softech AB, Västerås. Part time employment, with an equivalent total exceeding 2 years of full time work experience.

Publications

1. Thane H and Hansson H. *Testing Distributed Real-Time Systems.* Submitted for journal publication.
2. Thane H. and Hansson H. *Using Deterministic Replay for Debugging of Distributed Real-Time Systems*. In proceedings of the 12th Euromicro Conference on Real-Time Systems (ECRTS’00), Stockholm, June 2000.
3. Thane H. *Asterix the T-REX among real-time kernels. Timely, reliable, efficient and extraordinary.* Technical report. In preparation. Mälardalen Real-Time Research Centre, Mälardalen University, May 2000.
4. Thane H, and Wall A. *Formal and Probabilistic Arguments for Reuse and testing of Components in Safety-Critical Real-Time Systems.* Technical report. Mälardalen Real-Time Research Centre, Mälardalen University, March 2000.

5. Thane H. and Hansson H. *Handling Interrupts in Testing of Distributed Real-Time Systems*. In proc. Real-Time Computing Systems and Applications conference (RTCSA'99), Hong Kong, December 1999.
6. Thane H. and Hansson H. *Towards Systematic Testing of Distributed Real-Time Systems*. In proceedings of the 20th IEEE Real-Time Systems Symposium (RTSS'99), Phoenix, Arizona, December 1999.
7. Thane H. *Design for Deterministic Monitoring of Distributed Real-Time Systems.* Technical report, Mälardalen Real-Time Research Centre, Mälardalen University, November 1999.
8. Norström C., Sandström K., Mäki-Turja J., Hansson H., and Thane H. *Robusta Realtidssytem*. ***Book***. MRTC Press, September, 1999.
9. Thane H. and Hansson H. *Towards Deterministic Testing of Distributed Real-Time Systems.* In Swedish National Real-Time Conference SNART'99, August 1999.
10. Thane H. *Safety and Reliability of Software in Embedded Control Systems*. Licentiate thesis. TRITA-MMK 1997:17, ISSN 1400-1179, ISRN KTH/MMK/R--97/17--SE. Mechatronics Laboratory, the Royal Institute of Technology, S-100 44 Stockholm, Sweden, October 1997.
11. Thane H. and Norström C. *The Testability of Safety-Critical Real-Time Systems.* Technical report. Dept. computer engineering, Mälardalen University, September 1997.
12. Thane H. *Safe and Reliable Computer Control Systems - an Overview*. In proceedings of the 16th Int. Conference on Computer Safety, Reliability and Security (SAFECOMP'97), York, UK, September 1997.
13. Thane H. and Larsson M. *Scheduling Using Constraint Programming*. Technical report. Dept. computer engineering, Mälardalen University, June 1997.
14. Thane H. and Larsson M. *The Arbitrary Complexity of Software*. Research Report. Mechatronics Laboratory, the Royal Institute of Technology, S-100 44 Stockholm, Sweden, May 1997.
15. Eriksson C., Thane H. and Gustafsson M. *A Communication Protocol for Hard and Soft Real-Time Systems.* In the proceedings of the 8th Euromicro Real-Time Workshop, L'Aquila Italy, June 1996.
16. Thane, H. *Safe and Reliable Computer Control Systems - Concepts and Methods*. Research Report TRITA-MMK 1996:13, ISSN 1400-1179, ISRN KTH/MMK/R-96/13-SE. Mechatronics Laboratory, the Royal Institute of Technology, S-100 44 Stockholm, Sweden, 1996.
17. Eriksson C., Gustafsson M., Gustafsson J., Mäki-Turja J., Thane H., Sandström K., and Brorson E. *Real-TimeTalk a Framework for Object-Oriented Hard & Soft Real-Time Systems*. In proceedings of Workshop 18: Object-Oriented Real-Time Systems at OOPSLA,Texas,USA,October 1995.