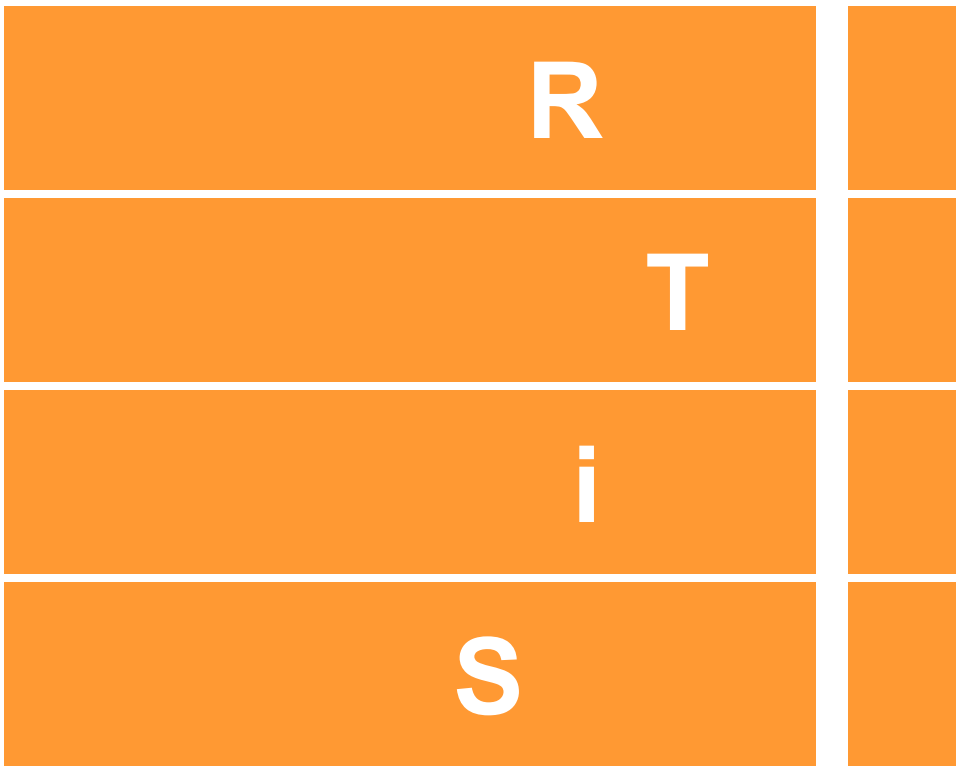
Real Time in Sweden 2001

Conference guide



SNART

Swedish National Real Time Association

Halmstad

21-22 August, 2001

SNART

The Swedish National Real-Time Association

Real-Time in Sweden 2001

Preface to the Real-Time in Sweden 2001 Conference Guide

Sweden has a very strong industry and high competence in the area of real-time systems, spanning applications in telecom, industrial automation, vehicles including aircraft, and medical equipment. The formation of the Swedish National Real-time Systems Association, SNART, in the early 90's was therefore a strategic and important step. It is important to say that real-time is only one distinguishing quality in the above mentioned applications. Other issues ranging from system safety, real-time control, low power, systems integration, verification etc. are strongly related and important, and fit well within the SNART umbrella.

SNART was formed with the overall aim of supporting and promoting Swedish real-time systems research. An important aspect of this is to provide a forum for information exchange open for all interested parties. In particular it is important to provide a bridge between different disciplines, and between industry and academia.

To this aim, this sixth biannual SNART conference has a program that differs both from traditional academic conferences and industrial fairs (and there are plenty of each category). In contrast, this conference emphasizes the overall picture rather than specific research projects, and has a program built around six general topics with plenty of room for personal and/or more specialized discussions, meetings, and seminars. To further emphasize the novel focus with respect to the earlier SNART conferences, a new name - Real-Time in Sweden - was suggested by Jonas Vasell and introduced for this conference. The goal is to provide system designers, researchers and others with an interest in real-time issues with information about important real-time problems and solutions, ongoing research and development efforts in Sweden, trends and future prospects, and who is doing what. This year, the conference topics include operating systems and middleware, applications: vehicular systems, programming, communication, system design, and safety-critical systems. Each topic will be presented from both an industrial and an academic perspective by experienced and renowned researchers and developers. Another difference with respect to traditional conferences is that the call for papers invited projects and research groups from academia and industry to make presentations of their activities.

Today, Swedish research in real-time systems and related areas has come a long way since SNART was formed. SNART paved way for ARTES - the national Swedish strategic research initiative in Real-Time Systems supported by the Swedish Foundation for Strategic Research (SSF). More than 100 graduate students are today linked to ARTES, which has been a major boost to academic research and education in the area. For the benefit of Sweden, we hope that the forum established by SNART and RTiS2001 will be succesful.

We are glad to note that we with the new conference format have managed to attract a wide range of contributions to the conference. The result is a rich conference program that provides a good overview of Real-Time in Sweden. A large number of speakers with different backgrounds from industry and academia have been invited. Some 20 contributions encompassing presentations of research groups, research projects, and industry activities were also received.

Organizing this conference would have been impossible without the help of many people. Special thanks go to Eva Nestius, Roland Grönroos, Carl Bergenhem, and Magnus Jonsson who all have helped with the practical arrangements; the topic coordinators who have put together excellent and varied topic presentation; and the program committee for its reviews of contributions and suggestions for program improvements.

We hope that this new conference concept will be interesting and provide a basis for future conferences, and that you will all enjoy your visit to Halmstad!

Sweden, August 6, 2001

Martin Törngren, Ola Redell, Jonas Vasell

SNART

Real-Time in Sweden 2001 programme

	2001-08-21			2001-08-22		
8:30-9:15	Registration, SNART coffe			Annual SNART meeting		
9:15-9:30	Introduction					
9:30-11:00	Topic 1 - Real-Time Operating Systems			Topic 4 - System Design: Towards System-on-Chip		
11:10 - 12:30	MRTC 1 MRTC 2		BoF	Session 1	Session 2	BoF
	HH (20)					
12:30 - 13:30	LUNCH			LUNCH		
13:30 - 15:00	Topic 2 - Applications: automotive systems			Topic 5 - Communication: keeping in touch in real time!		
15:10 - 15:50	DICOSMOS		BoF	SP	Gatespace 1 Gatespace 2	BoF
	ARTES			Data Respons		
16:00 - 17:30	Topic 3 - Programming: Software Development Programvaruutveckling			Topic 6 - Safety critical systems		
17:40 - 18:20	The Best Real-Time Master Thesis 2001	SIGURT	BoF	End of meeting (17:30-17:45)		
19:30	Dinner Tylöhus					

Topics

1. Real-Time Operating Systems
2. Applications: automotive systems
3. Programming: Software Development
4. System Design: Towards System-on-Chip
5. Communication: keeping in touch in real time!
6. Safety critical systems

Session 1

- Luis Alejandro Cortés et al. (LiTH), Verification of Embedded Systems using a Petri Net based Representation, (20 min)
- Thiemo Voigt (SICS) and Per Gunningberg (UU), Handling Persistent Connections in Overloaded Web Servers, (20 min)
- Håkan Sundell (CTH), Applications of Lock and Wait-free shared data structures to Real-Time Systems, (20 min)
- Radu Dobrin and Gerhard Fohler (MdH), Task Attribute Assignment of Fixed Priority Scheduled Tasks to Reenact Off-line Schedules, (20 min)

Session 2

- Anders Wall et al. (MdH), Product Line Architectures for Embedded Real-Time Systems, (20 min)
- Kristina Ahlström (CHALMERS), Next generation architecture for flight control systems, (20 min)
- Torbjörn Ekman et al. (LTH), Real-Time Java, (20 min)
- Aleksandra Zagorac et al. (LiTH), Software Development Tools for Embedded Databases in Embedded Real-Time Systems, (10 min + utställningsplats)
- Tobias Amnell et al., Software Synthesis Guaranteeing Timing Constraints, (10 min + utställningsplats)

Presentationer av forskningsgrupper samt projekt

- MRTC -1, Hans Hansson, (30 min)
- MRTC-2, Lennart Lindh, (30 min)
- Högskolan i Halmstad (HH), Bertil Svensson (20 min)
- DICOSMOS. Martin Törngren, (20 min)
- ARTES, (20 min)
- SP, Lars Strandén, Standards for Software in Safety-related Applications, (20 min)
- Data Respons, Realtidslösningar för Marin Stabilisering, (20 min)
- The OSGi platform and device access model, Presentation av OSGis modell för hantering av enheter och drivrutiner i en distribuerad och inbäddad miljö, Tommy Bohlin och Jan Sparud. (40 min)

The Best Real-Time Master Thesis 2001

- Winner 2001: Ulrik Hagström and Anders Kallerdahl (HH), "SIRTE - Switched Industrial Real-Time Ethernet"

- Honourable mention: Magnus Antonsson, Pernilla Hansson (CTH), "Modeling of Real-Time Systems in UML with Rational Rose and Rose Real-Time based on RUP"

Other issues

- SIGURT - "Undervisning i Realtidssystem". Något osäker programpunkt. Anordnas om intresse och initiativ finns.
- SNART annual meeting, previous meeting notes
- BOF - Birds of Feather sessions: there is extra rooms for spontaneous meetings.
- SNART coffe. There is coffebreaks planned and after the ARTES presentation will sandwiches be served.

Sessions chairmen

Day 1: Björn Lisper (Session 1), Jonas Vasell (DICOSMOS+ARTES), Martin Törngren (SNART-exjobb), SIGURT (?)
Day 2: Christer Norström (MRTC 1+2, HH), Ola Redell (Session 2), Ulf Gunneflo (SP + Data Respons), Jan Jonsson (Gatespace)

Utställningar - Demonstrationer - Posters

- Tobias Amnell et al., "Software Synthesis Guaranteeing Timing Constraints"
- Aleksandra Zagorac et al. (LiTH), "Software Development Tools for Embedded Databases in Embedded Real-Time Systems"
- De Jiu Chen (KTH), MARCH - Towards Systematic Architectural Design of Mechatronics Software Systems
- Ola Redell och Jad El-Khoury, AIDA II and PICADOR
- CR&T, Company poster
- Gatespace, The OSGi platform and device access model, "Presentation av OSGis modell för hantering av enheter och drivrutiner i en distribuerad och inbäddad miljö"

SNART

The Swedish National Real-Time Association

Real-Time in Sweden 2001

Organizers

General chair

- Jonas Vasell (Carlstedt Research & Technology/Halmstad University)

Program co-chairs

- Martin Törngren (KTH)
- Ola Redell (KTH)

Program committee

- Jakob Axelsson (Volvo)
- Åke Bromö (ABB)
- Ivica Crnkovic (Mälardalen University)
- Roland Grönroos (Uppsala University)
- Ulf Gunneflo (CR&T)
- Hans Hansson (Mälardalen University)
- Jörgen Hansson (Linköping University)
- Jan Jacobsson (SP)
- Jan Jonsson (Chalmers)
- Magnus Jonsson (Halmstad University)
- Lennart Lindh (Mälardalen University)
- Björn Lisper (Mälardalen University)
- Christer Norström (Mälardalen University)
- Anders Nyman (Ericsson)
- Ulf Olsson (Ericsson)
- Zebo Peng (Linköping University)
- Ola Redell (KTH)
- Anders Törne (CR&T)
- Martin Törngren (KTH)
- Jonas Vasell (CR&T/Halmstad University)
- Per-Arne Wiberg (Free2Move/Halmstad University)
- Karl-Erik Årzén (Lund University)

Real Time in Sweden 2001

On behalf of The Swedish National Real Time Association
the programme committee wants to express its sincere appreciation for the sponsorship

SNART

RTiS 2001 Topic 1

Real-Time Operating Systems

State of the art from a research perspective

Jörgen Hansson, LiU
The number of real-time operating systems being developed by companies and researchers has increased significantly over the last couple of years. The development emerges from the need for application-specific operating system, providing the necessary functionality with minimum cost with respect to system resources. This talk will include an overview of research problems, recent research advances, and developed research platforms. Further, we discuss the degree of flexibility in tailoring a real-time operating system for different types of applications.

The Design of Asterix RTOS

Henrik Thane, MDH
This talk describes a new real-time kernel, Asterix, which in a practical manner makes use of many of the recent advances made in the real-time systems research community. The basic ambition behind the development of the Asterix real-time kernel was to pack state-of the art research results into package such that it can be easily used and understood by people in the embedded systems industry. From an academic point of view the Asterix real-time kernel fulfills all the basic requirements necessary for facilitating different types of timing analyses. For a software designer this signifies that the Asterix real-time kernel has the means to satisfy engineering of real-time software in the same fashion as civil engineers make use of structural calculus when designing bridges or houses.

RTOS of the Next Generation

Jan Lindblad , ENEA
This talk reports on current industrial practice when developing real-time applications on top of real-time operating systems. We discuss the requirements that need to be satisfied by the operating system, and the technical challenges that the real-time designer is faced with when developing software for a real-time operating systems. In doing so, we outline some requirements and technical challenges that we believe need to be handled by the next generation of real-time operating systems.

Presentation of Mälardalen Real-Time Research Centre (MRTC) at Mälardalen University

Mälardalen Real-Time research Centre (MRTC) was initiated by a grant from the Swedish KK-foundation *(Stiftelsen för Kunskap och Kompetensutveckling)* to further develop the real-time research at Mälardalen University (MdH) in close co-operation with Swedish industry. As a result, a group of leading industries has joined the MRTC-effort by supporting industrial graduate students and participating in research projects. Strong support from Mälardalen University, the Swedish Foundation for Strategic Research (SSF; via the programme ARTES) and the Swedish National Board for Industrial and Technical Development (NUTEK/Vinnova) has enabled a fast build up of a focused research programme with a healthy balance between applied and fundamental research.

The research plan for MRTC is based on a three-pronged vision:

1. To provide state-of-the-art competence for industry.
2. To advance basic and applied research in relevant areas.
3. Education for engineers and researchers.

The advancement of these are mutually supportive, in that insights gained in one will guide the advancement in the others.

On a more technical level the guiding vision is to

provide engineers with substantially better tools and methods for the development of real-time computer systems and applications.

In this presentation we will introduce

- the overall organisation, the goals, the funding situation, and the pros and cons of building up of a research organisation at a small university,
- the education in real-time at undergraduate level, graduate level, and programmes for industrial active people, and
- three current research projects in the areas system design, computer architecture and computer science.

Contact person:
Hans Hansson, hans.hansson@mdh.se

Laboratory for Computing and Communication, Halmstad University

Brief Description of the Research Team

The laboratory for Computing and Communication at the school for Information Science, Computer and Electrical Engineering at Halmstad University is a constellation of computer engineers and computer scientists. It includes two professors, three associate professors and around 10 PhD students. The main concern of the team is within embedded systems. The staff at the laboratory is currently engaged in establishing a center for research and graduate studies on this field in close collaboration with industry that goes under the name of CERES.

The computer science staff is interested in the development of software tools in the form of domain specific languages that can assist the engineer in the development of embedded systems. These tools are meant to contribute with automatic controls, assistance in model transformations and connection to other tools like automatic theorem provers and model checkers.

Fields in the area of computer engineering which the lab are doing research on include low energy, high-performance, and special-purpose computer architectures, mainly with a focus on embedded systems. A lot of the research has been in the fields of parallel processing and real-time systems.

Research in the computer communication area is, in the lab, represented by both wireless and wired communication, but with a strong focus on embedded, industrial, and/or real-time systems. Examples of specific fields are: industrial real-time Ethernet; real-time, multi-hop, measurements/analysis, coding, and ARQ for wireless communication; time deterministic services; real-time multiple access for novel fiber-optic networks; wormhole networks; terabit routers and switches; interconnection networks for parallel/distributed computers; heterogeneous real-time services.

The laboratory has also responsibility for the areas of computing science, computer engineering and computer communication when it comes to the educational programs driven by the university. In this respect many of the courses that have been developed are permeated by real time issues. These courses include an undergraduate course on Real Time Programming and graduate courses on Distributed Real Time Systems, Parallel Computer Architectures and Real Time Communication.

SNART

RTiS 2001 Topic 2

Applications: automotive systems

This session deals with real-time systems from the viewpoint of automotive systems but also including a comparison with aircraft. The evolution of microelectronics, software, sensors and actuators, pave way for an unprecedented increase in software based functionality in automotive systems. Examples of new functionality being introduced into road vehicles include

- Closed loop engine control to reduce emissions and increase efficiency
- Advanced vehicle dynamics control including vehicle stability and autonomous driver assistance to improve safety and comfort
- X-by-wire functionality where for example the mechanical link between driver and steering wheels is replaced by a computer control system
- Electrical energy management systems to handle the increasing loads
- Intelligent transportation systems, where vehicles communicate with each other and the information infrastructure in order to increase the efficiency of the transportation system as a whole

The above examples clearly illustrate that the automotive industry is facing a paradigm shift where functionalities are being implemented using embedded computer control systems incorporating novel sensors and actuators. All this certainly implies a huge number of interesting research problems!

This session will provide several perspectives to automotive systems including

- an industrial perspective, describing current automotive electronic systems and complexity issues in future systems through a truck convoying case study. Presented by Jakob Axelsson , PhD, CR&T
- a research perspective, discussing research challenges, giving an overview of research areas and pinpointing architectural and model based design as a way towards meeting the complexity, dependability and time to market challenges. Presented by Martin Törngren, Ass. Prof, Mechatronics/KTH
- a tutorial on x-by-wire systems, introducing problems, design issues and some recent steer and brake by wire prototypes. Presented by Olov Bridal , PhD, Volvo Technological Development
- a panel session on the theme "Cost-efficient architectures for safety critical automotive systems - dreams or reality?"
- Within the panel session Prof. Jan Torin will contribute with experiences from digital flight control systems.

The scope of automotive applications is quite large and clearly spans most, if not all, of the other topics at RTiS. The session aims to provide links to the other topics as appropriate.

Updated Tuesday, 31-Jul-2001 16:09 by Roland Grönroos
e-mail: rolandg@it.uu.se
Location: http://www.docs.uu.se/snart/conference/rtis2001/topic2a.shtml

DICOSMOS - Distributed Control of Safety Critical Motion Systems

www.damek.kth.se/~mis/dicosmos/

DICOSMOS is funded by VINNOVA in the Complex Technical Systems/System Architecture program, and is carried out in cooperation between Mechatronics/KTH, Volvo Technological Development, Automatic Control/LTH, and Computer Engineering/CTH. The current project is running through 1999 - 2001, ending 2001-10-31. The project has a prehistory from 1993.

With relevance to the area of real-time systems, the DICOSMOS project is focussed on methods and techniques for architectural design of safety critical distributed control systems. In particular, the project has an interdisciplinary approach in the boarderland between automatic control and computer engineering. Up till now there has been a considerable gap between the automatic control and computer engineering disciplines; both in academic research and in industrial practice. DICOSMOS has over the year succesfully exploited this gap in the areas of real-time control, fault-tolerant communication, and in design methodologies. A list of publications and results from both the current project and the earlier phase is available from the project home page.

During the early design stages, the timing requirements are analysed and solutions from the control engineering perspective and from the computer engineering side need to be considered. To ensure dependable and cost-efficient solutions, it is essential that suitable information is exchanged between control and computer engineering during the developmen (fault avoidance). In addition, it is necessary to determine and exploit information exchange between the two levels during run-time. This can be used to obtain cost-efficient fault-detection and tolerance.

In the current DICOSMOS project a case study has been used as a common denominator for the project partners. The case study concerns modeling, simulation and analysis of an electrical brake and stability system for trucks. The application is a distributed real-time and safety-critical control system. The main focus with respect to functionality is on ABS and active yaw control functions. Within the case study a number of themes have been studied. The following are the ones most relevant to the RTiS conference.

- Architectural design looking at how to obtain a suitable computer system structure and functional allocation with respect to reliability, safety and modifiability [1-2]. In cooperation with the AIDA/PICADOR (ARTES funded) projects, a tool environment for architectural design is being developed [3].
- Timing problems referring to sampling period selection, jitter in period and feedback delays and transient errors. This topic was treated extensively in the earlier DICOSMOS project [4]. In the current project Quality of Service mechanisms are being investigated as a potential way of increasing the system robustness, [5]. As a basis for control design and studying the timing problems, a truck model for yaw and roll dynamics control was derived, and validated against real data [6]. Other work deals with delays in multirate systems [7].
- Real-time fault-tolerant communication and fail silent nodes [8-9]. This theme includes studies of communication system services that appropriately supports control applications, dealing for example with cost-efficient implementation of atomic broadcast. One related issue is were to place error detection mechanisms and, where and how to handle failures.

The project presentation will focus on the case study, the problem formulation and initial results.

Martin Törngren Project leader, DICOSMOS
For more information on DICOSMOS refer to http://www.damek.kth.se/~mis/dicosmos/

References:

[1] Sanfridson Martin, Claesson Vilgot, Gäfvert Magnus. *Investigation and requirements of a computer control system in a heavy-duty truck*. Technical report, Mechatronics Lab, Department of Machine Design, Royal Inst. of Technology, Stockholm. TRITA-MMK 2000:5, ISSN 1400-1179, ISRN KTH/MMK--00/5--SE.

[2] Claesson Vilgot, Gäfvert Magnus, Sanfridson Martin. *Proposal for a distributed computer control system n heavy duty truck*s. Report no. 00-16, Dept. of Computer Engineering, Chalmers Univ. of Technology, Göteborg. Scheduled for autumn 2000.

[3] Törngren Martin, El-khoury Jad, Sanfridson Martin, Redell Ola. *Modelling and Simulation of Distributed Real-time Control Systems*. Technical report, Mechatronics Lab, Department of Machine Design, Royal Inst. of Technology, Stockholm. TRITA-MMK 2000:XX, ISSN 1400-1179, ISRN KTH/MMK--00/XX--SE. Scheduled for autumn 2000.

[4] Törngren, M. and M. Sanfridson, editors (1998): Research problem formulations in the DICOSMOS project. TRITA MMK 1998:20, ISSN 1400-1179. ISRN KTH/MMK--98/20--SE.

[5] Sanfridson Martin. *Timing Problems in Distributed Control*. Licentiate Thesis. Mechatronics Lab, Department of Machine Design, Royal Inst. of Technology, Stockholm. TRITA-MMK 2000:14, ISSN 1400-1179, ISRN KTH/MMK--00/14--SE.

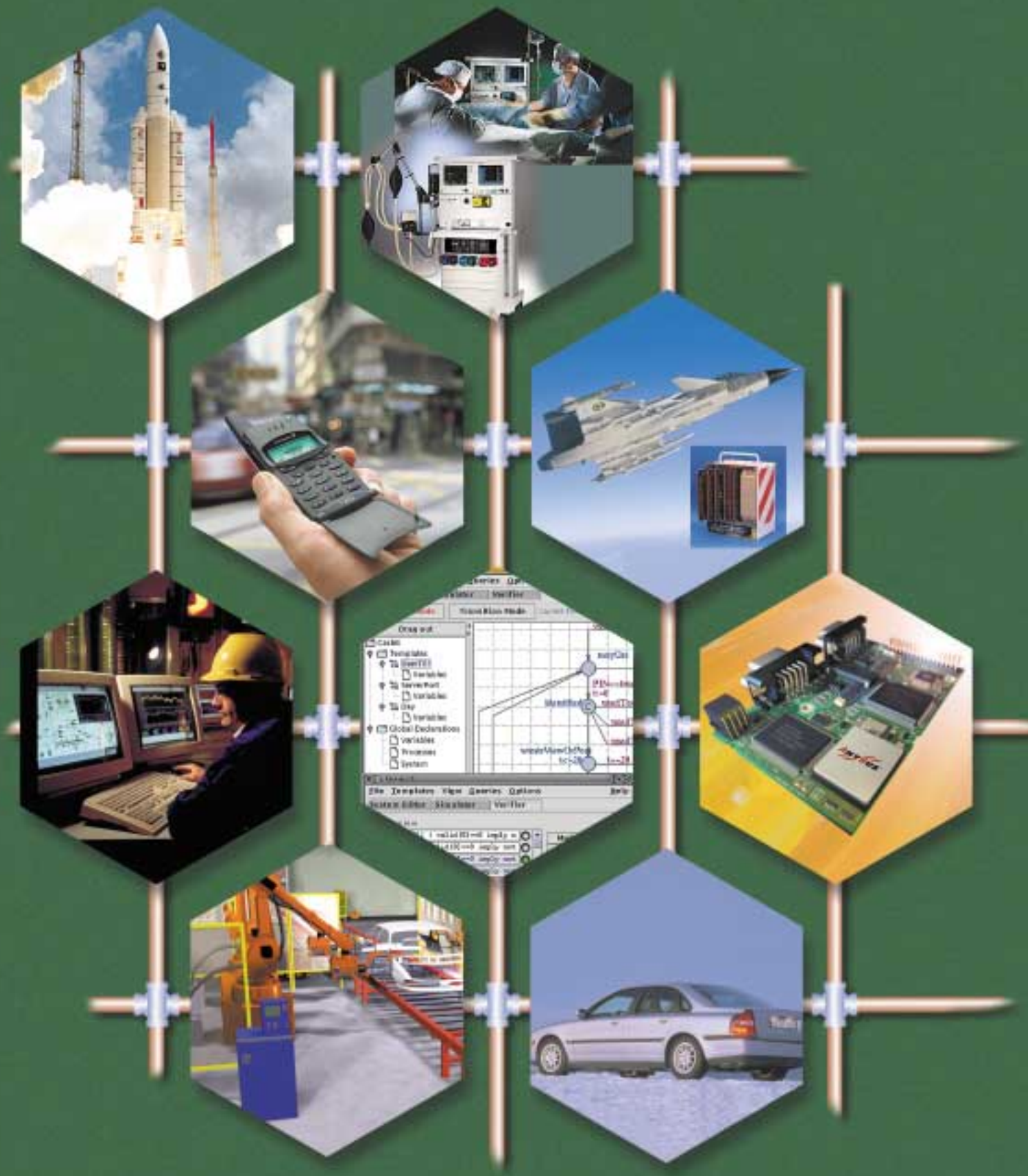
[6] Gäfvert Magnus, Sanfridson Martin, Claesson Vilgot. *Truck Model for Yaw and Roll Dynamics Control*. Department of Automatic Control, Lund Institute of technology, Sweden, 2000, ISRN LUTFD2/TFRT--7588--SE

[7] Wittenmark, Björn. Sample-induced delays in synchronous multirate systems. Accepted for *European Control Conference*, Porto, Portugal, September 4-7, 2001.

[8] Claesson Vilgot. *Cost-effective communication services for applications in distributed time triggered real-time systems*. Licentiate. Thesis 316L, Dept. of Computer Engineering, Chalmers Univ. of Technology, Göteborg 1999.

[9] Askerdal Örjan. *Design and Evaluation Techniques for Detection and Coverage Estimation of Low-Level Errors*. Licentiate. Thesis, Department of Computer Engineering, Chalmers University of Technology, Göteborg, Sweden 2000.

Take advantage of ARTES



ARTES

A network for Real-Time research and graduate Education in Sweden

Real-time systems

An enabling technology with numerous applications

Real-time systems are computer systems that sense their environment and directly influence it through actions.

Real-time systems must not only choose appropriate actions, but also choose them at appropriate times. Most real-time systems are embedded in products. For instance, an autonomous vehicle will have an embedded computer-based control system that has to respond in time to avoid collisions. Real-time computing is not about building "fast" systems; it is about building systems that are predictably "fast enough" to interact with their environments in well specified ways.

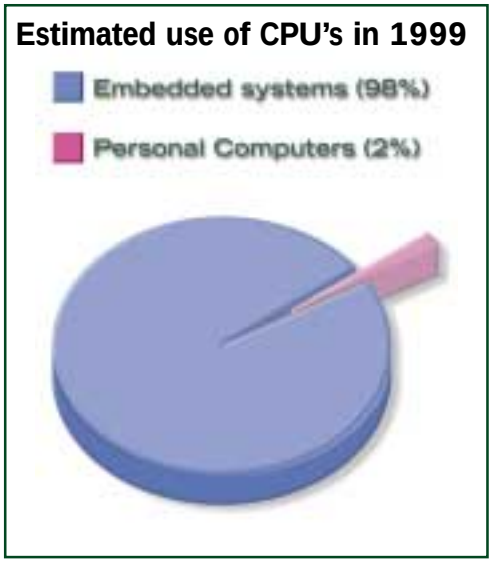
Real-time systems are embedded in a multitude of applications and products, in areas such as multimedia, telecommunications, robotics, process control, flexible manufacturing, avionics, vehicular systems, air-traffic control, nuclear power plants, medical equipment and defence applications. In fact, the vast majority of processors used are in embedded systems.

Courtesy of Husqvarna AB

Competencies vital to Swedish industry

Real-time computing plays a key role in many sectors of Swedish industry. For example, the automotive manufacturers can only stay competitive if they incorporate state-of-the-art real-time systems into their vehicles.
In the future, distributed real-time control systems will replace and enhance many of the conventional control systems in cars, making them safer and more efficient.

In order to fully utilise distributed real-time systems, a number of significant research challenges must be addressed, e.g. precise real-time response (to the microsecond), fault tolerance under strict timing requirements, serviceability and testability, all under competitive price pressure.



Reference: Jim Turley, Microprocessor Report

A multidisciplinary research area

Developing real-time systems demands knowledge of and contacts with a number of research disciplines, including automatic control, computer science, computer engineering and electrical engineering.

ARTES' mission is to facilitate the transition of real-time technology between academy and Swedish industry and to stimulate graduate education, thereby helping industry solve the critical problem of finding qualified personnel.

"ARTES offers an enhancement of research and education targeting computer systems integrated in products and industrial processes.
ARTES network for research and industry will lead to a stronger Swedish competence in technical development of great importance for our manufacturing industry."

Ingvar Carlsson,
Chairman, Swedish Foundation for Strategic Research

This is ARTES

Our twofold vision:

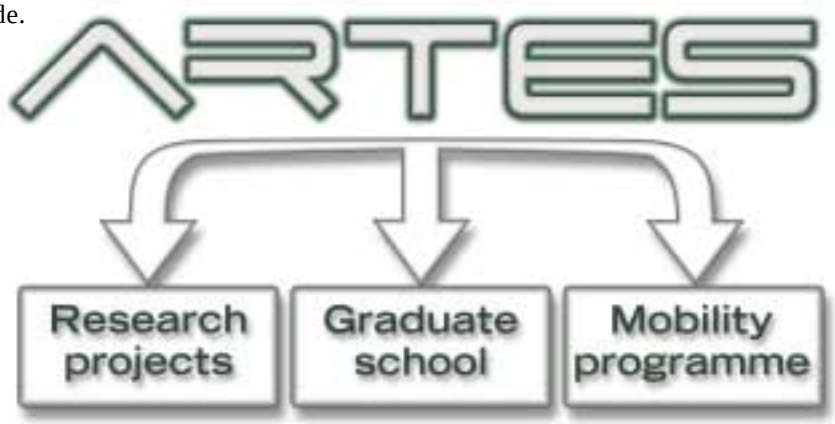
To transfer knowledge and competence to Swedish industry that will allow it to first utilise the latest achievements in real-time systems design

To reduce lead times for designing and modifying real-time systems by an order of magnitude by the year 2005

A strategic research initiative

ARTES is a Swedish national research initiative in real-time systems, supported by the Swedish Foundation for Strategic Research (SSF), which funds the programme with MSEK 88. ARTES was formed in 1997, following an initiative from The Swedish National Real-Time Association (SNART). ARTES is organised as a research programme at Uppsala University and forms a national network of academic and industrial groups, with the ambition to strengthen the real-time systems competence nation-wide. The main focus of ARTES is on graduate education and co-operation between industry and academia.

The ARTES programme consists of three interrelated and mutually supportive sub-programmes.



"We live in real-time. Most future products will have their main functionality in the form of built-in electronic systems. The demands for interaction with other products and with human users mean that all these systems must be designed for real-time performance."

Bernt Ericson,
Vice President, Research and Innovations, Ericsson

Research projects

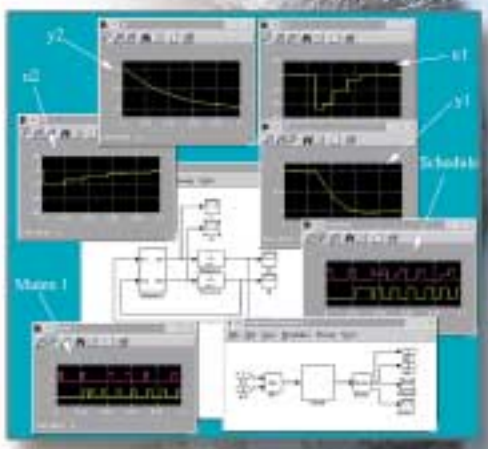
www.artes.uu.se/project

Co-operation between academia and industry is a vital ingredient in all ARTES projects. Equally important is to provide relevant research problems and environments for graduate students.

A typical project considers several research areas, but is focused on one application area. Preferably, the project should apply research results to real industrial problems, or develop results conforming to industrial requirements.

Currently there are more than twenty ARTES projects in a wide variety of areas. Here are just a few examples:

Integrated control and scheduling



Most real-time systems are control systems. This project, conducted at Lund Institute of Technology, focuses on combining control theory and scheduling theory to achieve higher CPU utilisation and better control performance.

The theoretical results will be implemented in tools that can be used in industrial applications. One such tool is a simulator that makes it possible to study the signals for control and measurement, as well as the schedule and the allocation of shared resources in the computer system.

The industrial partners in the project include Sigma Exallon Systems and DDA Consulting.

Project leaders: Karl-Erik Årzén and Klas Nilsson, Lund Institute of Technology

System architecture for mechatronics systems

The MARCH (Mechatronics ARCHitecture) project explores design methodologies and develops a system architecture framework suitable for mechatronics systems controlled by an embedded distributed computer system. SAAB Automobile AB is one of the industrial partners.

As an example of an application, the project develops a system architecture for walking robots, with a case study being conducted at the Centre for Autonomous Systems at the Royal Institute of Technology (KTH) in Stockholm. The objective is to provide a robust and agile locomotion platform suitable for difficult terrain.

Project leaders: Martin Törngren and Jan Wikander, KTH

A tool environment for the development of embedded systems

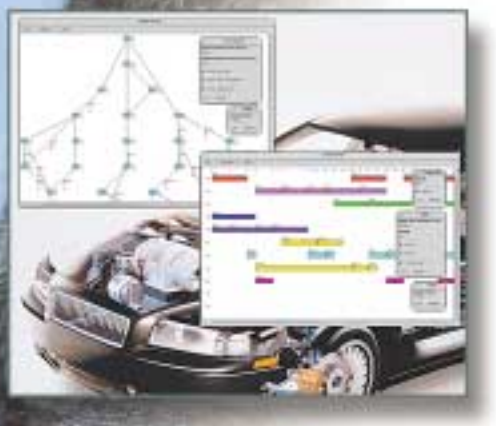
UPPAAL is a tool box for graphic modelling, debugging, validation and verification of real-time embedded systems, developed jointly by Uppsala University and Aalborg University in Denmark. It has been successfully applied all over the world in research, education and industrial case studies.

The vision of the research group is to extend UPPAAL into a coherent tool environment supporting each step of the system development process from requirements specification to testing.

Two Ph.D. projects are funded by ARTES.
Volvo Technological Development Corporation and Mecel AB are the industrial partners.

Project leader: Wang Yi, Uppsala University and Mälardalen University

Hardware - software co-design of real-time systems



Modern real-time systems are built from carefully optimized hardware and software components. As the complexity of such systems increases new design techniques that can address the concurrent development of hardware and software components (hardware/software co-design) are required.

The aim of this project, conducted at the Embedded Systems Laboratory at Linköping University, is to develop a real-time design environment consisting of a set of integrated design methods and tools that allow designers to quickly explore the design space and to produce optimized implementations. One of the application areas is automotive electronics, with Volvo Technological Development Corporation as industrial partner.

Project leaders: Zebo Peng and Petru Eles, Linköping University

Making VR appear real with multiprocessors

Virtual reality (VR) technology is opening up several new applications. Design alternatives (e.g. in car design) can quickly be prototyped and evaluated, risky training sessions (e.g. in surgical training) can be avoided, and new forms of education are made possible. Realistic VR demands high computing capacity, however.

At Chalmers University of Technology in Göteborg, algorithms central to VR are boosted using off-the-shelf multiprocessors. Despite their high computing power, a key challenge is to make applications exploit better the inherent parallelism. Parallelisation techniques for animated ray tracing are under investigation. The work is carried out in collaboration with ABB Digital Plant Technologies AB.

Project leader: Per Stenström, Chalmers

The VR project is a part of PAMP, a sub-programme within ARTES. PAMP (Performance-demanding Applications on MultiProcessors) develops design methods, tools and generic building blocks to reduce the time-to-market of industrial high-performance real-time applications using multiprocessor technology.

The ARTES graduate school

A real-time graduate student is a Ph.D. or licentiate student at a Swedish university with a thesis subject in computer science, computer engineering, computer systems, industrial control systems, mechatronics, automatic control or real-time systems. He or she is well motivated to work in co-operation with industry during his/her education and has an ambition to eventually make a career in industry. He/she contributes to ARTES' activities by making reports available on the Internet and by actively participating in ARTES events.

More than 70 students are already part of ARTES' real-time student network and ARTES offers them:

- ARTES summer school - an annual forum for discussions, contacts between students and industry representatives and in-depth tutorials by invited world-leading scientists
- ARTES support for developing and giving graduate courses in real-time subjects
- ARTES support to scientifically challenging research projects with industrial relevance
- ARTES support for developing real-time curricula at graduate as well as undergraduate level (partly in co-operation with SNART)

Mobility programme

In order to increase the interaction between industry and academy and the contacts with internationally leading research groups, the ARTES mobility programme provides support to facilitate personal mobility.

- International experts can take part in a research project.
- Industrial engineers/researchers can work in a research project at an academic node.
- ARTES graduate students can spend time at high-ranking international institutions or at Swedish companies as an integrated part of the research project.

"During the autumn of 1998, I spent three weeks at Volvo Technological Development Corporation, one of the industrial partners in our project. The visit was very successful in several ways. First, it helped me to better understand the automotive electronics field, which is one of the application areas of our research. Secondly, we were able to select a relevant case study for our project and, finally, it allowed me to identify other interesting research problems."

Paul Pop, graduate student at Linköping University

Industry seminars

If your company is actively developing real-time systems we suggest that you let ARTES visit you for a mutual exchange of experiences.

A real-time seminar is typically a half-day to a full-day meeting aimed at establishing contact and exchanging ideas between real-time scientists and practitioners.

You can choose to invite one or more specific ARTES projects.
Researchers working with these projects will then come and talk about their research and the latest developments in the real-time area. Your own researchers/engineers can outline their current work and any problems that the academic researchers can help them solve, or that might be suitable for inclusion in current or future ARTES projects.

This is an opportunity not only to get access to some of the latest research achievements, but also to influence the future direction of research, making it even more certain that your company will benefit from ARTES.

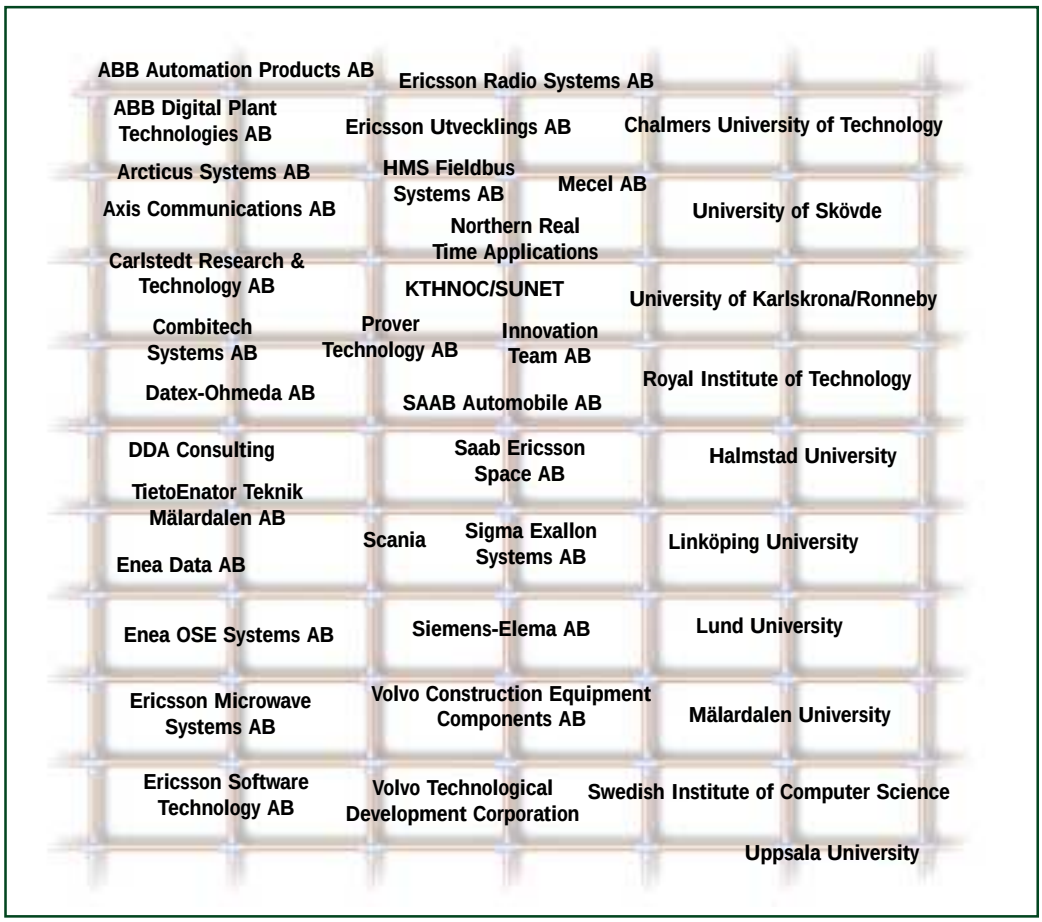
Information about the ARTES researchers and projects is available at

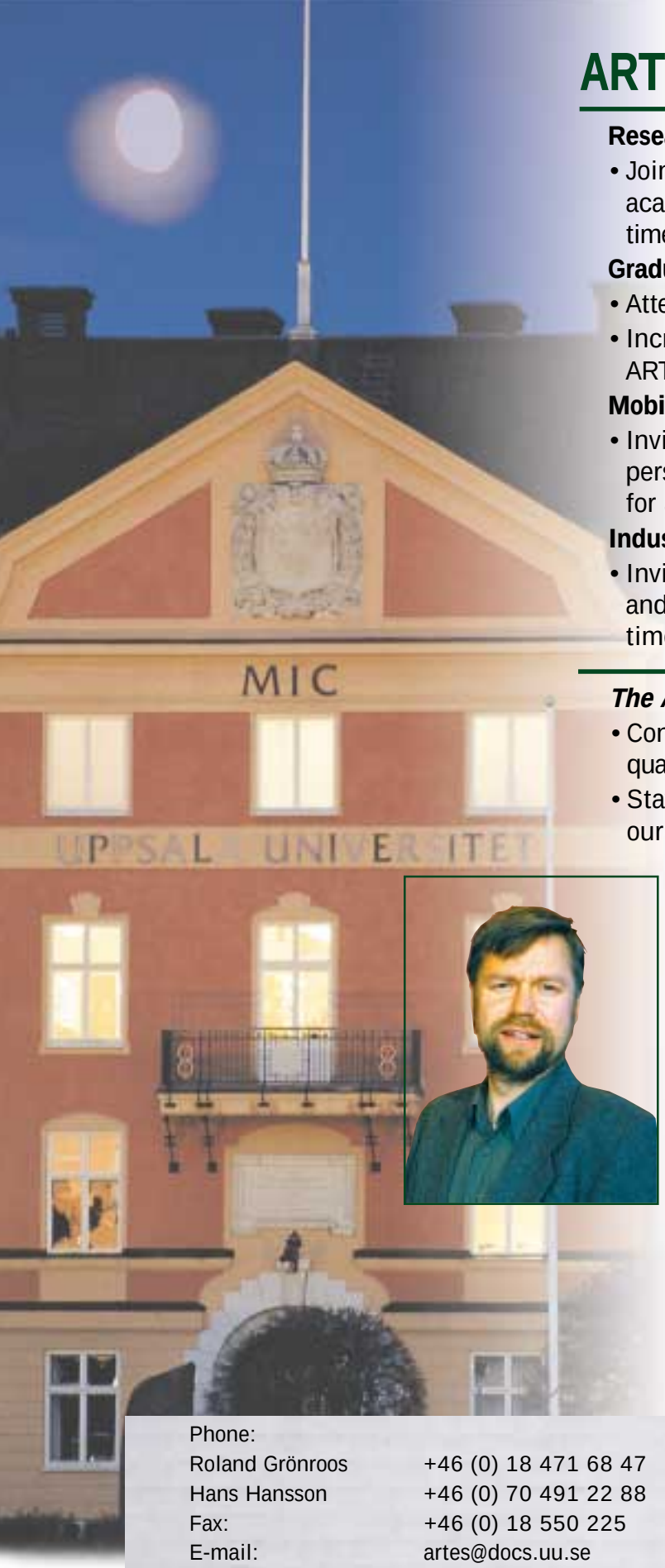
www.artes.uu.se/project

Please feel free to contact us for further information and to discuss the time for and contents of your ARTES seminar.

The ARTES network

The following companies and universities are currently involved in ARTES research projects.





ARTES is a valuable partner

Research projects

- Join this co-operation between industry and academic research and help develop the real-time areas of interest to your company.

Graduate school

- Attend the ARTES summer school to get in touch.
- Increase your competence by attending an ARTES course.

Mobility programme

- Invite a Ph.D. student to your company or swap personnel with an academic research institution for a period.

Industry seminars

- Invite academic researchers to exchange ideas and to discuss how you can co-operate in real-time systems development.

The ARTES Network

- Consider recruiting excellent personnel to highly qualified positions.
- Stay informed by adding your e-mail address to our list of interested persons.

"Real-time systems are of strategic importance to Swedish industry, and Sweden is a leading nation in real-time research. Join the ARTES network, and you get access to top researchers and their latest findings!"

Hans Hansson, Programme Director

Contact ARTES:

www.artes.uu.se

Phone:
Roland Grönroos +46 (0) 18 471 68 47
Hans Hansson +46 (0) 70 491 22 88
Fax: +46 (0) 18 550 225
E-mail: artes@docs.uu.se

ARTES
Box 325
SE-751 05 UPPSALA
Sweden

ARTES
A network for Real-Time research and graduate Education in Sweden

SNART

RTiS 2001 Topic 3

Programming: Software Development

Björn Lisper, MdH
A trend in real-time programming is towards methods that use specifications: code generation from a system- or functional specification, or programming languages with modern abstraction mechanisms. Another trend is the integration of different models for programming and specification to cover all aspects. Interesting questions are: how integrate, how ensure the functionality. How meet the contradictory demands for safety, predictable performance, resource lenience, and low costs for development and maintenance?

Modeling real-time systems with UML

Morgan Björkander , Telelogic
This talk provides a brief overview of the work being performed within the OMG on UML and its real-time characteristics. It specifically focuses on the next generation of UML (UML 2.0) and the first real-time UML profile being defined, which deals with capturing characteristics related to performance, scheduling, and time.

Biography : Morgan Björkander is working as methods specialist at Telelogic, where he has been employed since 1993. He received his MSc in Computer Science from the Lund Insitute of Technology, and has been involved in several European projects on distributed systems. During the last few years he has been the Telelogic representative within the OMG, where he focuses on the standardization of UML including its real-time capabilities.

Panel debate

Panel : Klas Nilsson (LTH), Inga-Lill Bratteby-Ribbing (FMV), Martin Törngren (KTH), Henrik Leerberg (IAR Systems)

SNART

SNART Awards

The Best Real-Time Master Thesis 2001

From 1997 a prize for the Best Master or Final Year Thesis is awarded at the conference/seminar. The prize is 2000 SEK + travel expences to and conference fee at Real-Time in Sweden 2001.

Best Master or Final Year Thesis Awards

2001 Winner 2001
Ulrik Hagström and Anders Kallerdahl
SIRTE - Switched Industrial Real-Time Ethernet
Abstract, Extended Abstract

Honourable mentions
Magnus Antonsson, Pernilla Hansson
Modeling of Real-Time Systems in UML with Rational Rose and Rose Real-Time based on RUP , Abstract

2000
Erik Wyke
Investigation of models for real-time systems: AIDA through UML and ROOM
Erik Wyke´s studied the very important problem of modelling embedded real-time systems. The thesis give an interresting and relevant survey of real-time system models. There are specific discussions of UML and ROOM, and how they can be used to support development of real-time systems. The studies were carried out at The Royal Institute of Technology (KTH), Stockholm
The thesis is available as Pdf.

Honourable mentions
Due to the even judgements, SNART has decided to honourably mention
Alejandro Garcia, Lisbeth Johansson, and Mattias Weckstén
for the study
Real-Time Services in Myrinet Based Clusters of PCs
carried out at Högskolan i Halmstad.
The thesis is available as Pdf.

and

Cinzia Foglietta
for the study
Assured Selection -- A Relaxed Concurrency Control Mechanism
carried out at Linköpings Tekniska Högskola.
The thesis is available as Pdf.

1999
Björn Andersson
Adaption of Time-Sensitive Tasks on Shared Memory Multiprocessors - A Framework Suggestion

The thesis gives a framework for the design of real-time systems on multiprocessors using a negotiator facility. The work is theoretically well-founded and an implementation has been made to test the approach. A well written report
The studies were carried out at Computer Systems, Chalmers.
The thesis is available as Postscript or Pdf. An abstract is also available.

1998
Jakob Engblom
Worst-Case Execution Time Analysis for Optimized Code

The thesis gives a very good description of worst case execution time analysis for optimized code, software analysis tool and ideas how to combine the tool with a compiler. The structure and description of the field is excellent. Also, the conclusions and future work is a very good start point for Jakob Engblomís licentiate work. Combining the analysis tool and the compiler is a new approach and can give the market a strong tool in the future.
Thestudies were carried out at DoCS (ASTEC), UU, Uppsala.
The thesis is available as Postscript or Pdf. An abstract is also available.

1997

Patrik Lantto
Requirement Analysis and Implementation of a Real-Time System
Handling a Large Amount of Data
(in swedish)

The work is based on a concrete industrial problem - data acquisition in a hard real-time system. An excellent requirement analysis is made and scheduling theory is applied. The thesis also contains a prototype implementation of an optimal priority assignment scheduler. The report is well written and the work should be of general industrial interest.
The studies were carried out at IDA,LiTH and OPQ Systems AB, Linköping.
The thesis is available as Postscript or Pdf (in Swedish). An abstract in English is also available.

Industrial Real-Time Ethernet

Ulrik Hagström, Magnus Jonsson, and Anders Kallerdahl

Preface

This project is a Master' s degree thesis and concludes the program Master of Science in Computer Systems Engineering with a profile in Innovation and Entrepreneurship at the School of Information Science, Computer and Electrical Engineering (IDE-section) at Halmstad University. The project was initiated in January 2000 and was concluded in January 2001. The project was made at the Halmstad University campus in collaboration with HMS Industrial Networks.

This report is part of a masters thesis report and has been written as an overview of the final thesis. The full report can be obtained by contacting Anders Kallerdahl (see below for contact address).

Introduction

Ethernet networks have not been suitable for industrial environments due to the nature of the undeterministic handling of communication-traffic. The traffic in, e.g., an industrial fieldbus network is usually time-constrained and requires therefore real-time services that Ethernet not has been able to provide. Today the prices of Ethernet hardware have fallen, transmission speed has increased, and the Ethernet standard (IEEE 802.3) has been extended with new features such as full duplex, flow control etc. Taking advantage of the mentioned progress, we have now developed a network concept (including novel methods) for which sophisticated support for real-time traffic is offered. The network includes an Ethernet switch with some specific hardware and software, while end nodes is only complemented with some additional software. We call our developed network concept Industrial Real-Time Ethernet. Significant features of the Industrial Real-Time Ethernet include:

- Real-time support for cyclic traffic with cycle duration as short as ca 1 ms.
- Worst-case latency down to ca 1 ms (can be shorter than the cycle duration).
- Dynamic configuration/set-up of real-time channels with guaranteed performance (Quality of Service; QoS). Static configuration at system design/start-up is also possible.
- Transparent support for reliable UDP/IP traffic, such as for standard TCP/IP but with much better timing properties.
- High transmission capacity (switched 100 Mbit/s full duplex).
- Clock synchronisation which can be utilised by, e.g., applications and/or operating systems.
- Remote maintenance of industrial applications via Internet.
- Support for multiple logical real-time channels from each node.

- Eliminates use of traditional complex industrial communication systems.
- Real-time support for standard UDP/IP Internet traffic.
- Support for co-existing non-real-time standard TCP/IP and UDP/IP Internet traffic without disturbing the real-time traffic.
- In addition to switched networks, Industrial Real-Time Ethernet can be extended with some support for real-time traffic over a shared-medium topology too.
- Real-time support can be given to applications without any modifications of them. Using an Industrial Real-Time Ethernet API (Application Programming Interface) will, however, give the opportunity of setting up and specify real-time channels more dynamically.
- Non-real-time nodes (nodes without Industrial Real-Time Ethernet drivers) can co-exist in the network without disturbing the real-time traffic.

In summary, we offer better support for industrial real-time traffic over Ethernet compared to existing solutions on the market and published research results. The solution brings cost advantages by, e.g., the use of cheap standard Ethernet mass-market components, remote maintenance, and coexistence of maintenance and real-time traffic.

The proposed solution

Industrial Real-Time Ethernet is based on an 100 Mbit/s Ethernet switch supporting hard real-time traffic with bit rate and latency guarantee, i.e. a predefined amount of data delivered before a deadline. The guarantee is uphold by real-time channels, which can be created and closed dynamically. A real-time channel can be considered as a virtual wired connection between two nodes connected to the switch, with a guaranteed bit rate and a guaranteed maximum latency. The nodes, which are connected to the switch, can either be real-time nodes or non real-time nodes depending on which level quality of service is required. Non real-time nodes are nodes without Industrial Real-Time Ethernet drivers.

In the proposed solution there are no modifications of the Ethernet hardware on the network interface cards. The real-time communication support in the nodes and the switch is handled by software added between the network layer and the link layer, in the OSI reference model. Some additional hardware in the switch, compared to a standard Ethernet switch, might also be needed. The switch has the overall responsibility both for set-up of real-time channels and for online control of packets passing through the switch. The software in the end nodes will include API and simple regulation of the traffic passing through the protocol suite and out of the node.

To mention some details of the solution, it is based on, e.g., cyclic resource allocation with relaxed synchronisation requirements, earliest deadline first algorithm, and control messages with synchronisation, acknowledgement and access control information. More details on the proposed solution can be given by the inventors (see the end of this document for contact information).

Vision

Industrial Real-Time Ethernet has a broad target market including application areas such as: (*i*) industrial communication industry with products used in, e.g., manufacturing industries and where our partner, HMS (HMS Industrial Networks), is one of the world's leading companies in the fieldbus technology sphere, (*ii*) communication in embedded systems like radar signal processing systems for which for which we have good contacts in one of the world leaders (Ericsson Microwave Systems), (*iii*) broadband access networks, (*iv*) traditional local area networks (LANs) where, e.g., enhanced support for multimedia traffic is needed, and (*v*) storage area networks (SANs) for the hot area of distributed storage. The work this far has been focused on the first two application areas (for which we have got very good indications of market interest from our partner and contacts) but we see no reason to exclude the other application areas since they have very similar demands of real-time traffic support.

Industrial Ethernet (www.industrialethernet.com) is an example of an organisation that is adapting Ethernet hardware to the industrial environment. Despite of the recent trends towards high-speed networks such as ATM, the predominant architecture for local area networks is still based on Ethernet. The standardisation work on industrial Ethernet, where HMS is a founding member of IndustrialEthernet.com and board member of IAONA (Industrial Automation Open Networking Alliance), has, however, not lead to sophisticated support for real-time traffic such as in Industrial Real-Time Ethernet.

There has been much discussion recently regarding the applicability of using Ethernet at various levels in industrial applications. Since Ethernet is so prevalent and frequently used in the office it would be natural to use Ethernet in the industrial plant-floor in industrial networks. Eric Byres, Artemis Industrial Networking, January 1998, quoted in Intech Magazine: "I think we may see Ethernet completely take over the industrial bus and network market in the next ten years". This is one reason that the development of Industrial Real-Time Ethernet was started.

Currently existing field-busses on the market have a maximum bit rate of 12 Mbit/s (Profibus with advanced transceivers) but the majority of the field-busses have bit rates from 200 kbit/s to 500 kbit/s. Industrial Real-Time Ethernet will use 100 Mbit/s due to the fact that future industrial systems require higher bit rates to allow the network to connect to the outside world (Intranet/Internet) and more complex systems. The connection to intranet/Internet makes, e.g., web based maintenance possible to coexist with the real-time traffic which is highly desirable. It will be possible to browse an industrial computer via the Internet for status control or for software updating and at the same time guarantee that this will not hazard the important real-time traffic.

There is a rapid increase of the need of online storage, such as network connected hard discs. Storage of data through a network needs reliable communication. Future industry trends says that future capital investments in storage will be 75 % of all capital investments in corporate data centres ["Network attached storage architecture", *Communications of the ACM*, Nov. 2000]. More market advantages with Industrial Real-Time Ethernet are treated in the introduction.

Our vision of a company based on Industrial Real-Time Ethernet, makes its money on selling the Ethernet switches with real-time support. Software drivers will probably be

shipped free of charge. Our partner in the project, HMS, has an established world-wide sales organisation which advantageously can be used.

Current state of the project

This project has been running as a master' s thesis work at the University ofHalmstad involving two master degree students, Ulrik Hagström and Anders Kallerdahl, and a supervisor, Ph.D. Magnus Jonsson. This project started in January 2000 and was finished in January 2001.
The goal of this project was to develop a specification for real-time communication over Ethernet via a switched network without modifying any Ethernet hardware in the end nodes. The reason not to modify the Ethernet hardware in the end nodes is that the market is conservative and in order to support inclusion of existing systems.

The status of the project is that a specification of the switch and nodes exist. The specification includes algorithms handling the scheduling of the communication and protocols for, e.g., channel set-up and error-handling. No hardware, software or simulation implementation has been done. Instead the main focus have been to make the proposal as generally applicable as possible and to get good knowledge of the state of the art in the area.

The first stage of future work will primary be software development and construction of a hardware demonstrator. We believe that the hardware can be developed based on standard components available on the market. The next stage will be to get funding for commercialisation.

Rewards

- ✍ This project has been chosen by Chalmers School of entrepreneurship to be a masters thesis project as this project was seen as a promising idea to apply for a patent and build a company on.
- ✍ This project has been rewarded with the best masters project award of 2000 by the company Innovation Team. (www.innovationteam.se).
- ✍ This project has been rewarded as the best master thesis by SNART (Swedish National Real-Time Association) 2001. (http://www.docs.uu.se/snart/)

Supervisor

Magnus Jonsson, Ph.D., Associate Professor (Universitetslektor)

School of Information Science, Computer and Electrical Engineering

Students

Ulrik Hagström, Anders Kallerdahl

Contact address

Anders Kallerdahl
Smyckegatan 4, 426 50 Västra Frölunda
phone: 031-476101 mobile: 0708-366353
e-mail: anderskallerdahl@hotmail.com

Modeling of Real-Time Systems in UML with Rational Rose and Rose Real-Time based on RUP

Magnus Antonsson and Pernilla Hansson, Chalmers

Abstract

In software development, using UML for modeling of real-time systems is a fairly new area. There are a lot of different theories and reports in this area and UML is in a development phase when it comes to modeling of real-time systems.

This report provides a summary of how and why UML and UML-RT can be used for modeling of real-time systems. It also includes a summary of the advantages and disadvantages of Rational Rose and Rose-RT concerning modeling of real-time systems.

This report also provides modifications that can be done to RUP to make the process better suited for development of real-time systems. This includes among others the use of state machines to capture the concurrency within and among Use Cases.

Sammanfattning

Att använda UML för modellering av real-tidssystem är ett relativt nytt område. Det finns flera olika teorier och rapporter som tar upp detta och UML är i en utvecklingsfas när det kommer till modellering av realtids system.

Denna rapport ger en sammanfattning av hur och varför UML och UML-RT kan användas för att modellera real-tidssystem. Den inkluderar också en sammanställning av de fördelar och nackdelar som verktygen Rational Rose och Rose-RT har när det gäller modellering av real-tidssystem.

Denna rapport tillhandahåller också förändringar som kan göras på RUP för att göra processen bättre lämpad för modellering av real-tidssystem. Detta inkluderar bland annat användandet av tillståndsmaskiner för att fånga parallellitet inom och mellan Användningsfall.

SNART

SIGURT

Snart's Interest Group on Undergraduate education in Real-Time systems

or

Snarts Intressegrupp för GrundUtbildning i RealTidssystem

E-mail list snart-sigurt@docs.uu.se to all SIGURT Members

Notes and slides from the first meeting 99-10-19 at Chalmers as pdf-files.

Real-Time Courses in Sweden

Updated Tuesday, 22-Feb-2000 15:29 by Roland Grönroos
e-mail: rolandg@it.uu.se
Location: http://www.docs.uu.se/snart/education/sigurt/

Årsmötesprotokoll

SNART 2000-08-22

Årsmötet hölls vid Sundbyholms slott, Eskilstuna, arrangör ARTES/KTH.

§1. Mötet öppnades.

§2. Martin Törngren valdes till mötesordförande och Anders Törne valdes till mötessekreterare.

§3. Till justeringsmän valdes Jan Jonsson och Karl-Erik Årzén.

§4. Agendan för årsmötet godkändes.

§5. Styrelsens verksamhetsberättelse presenterades av ordf. I bilaga finns verksamhetsberättelse och bokslut.

§6. Skriven revisorsberättelse inkommer till ordförande Martin Törngren.

§7. Styrelsen beviljades ansvarsfrihet för det gångna verksamhetsåret under under förutsättning att revisionsberättelsen godkänner de ekonomiska transaktionerna.

§8. Val av styrelse

Till styrelse valdes:

Martin Törngren, KTH, ordförande, Zebo Peng, Linköping, Jan Jonsson, Chalmers, Christer Norström, MdH, Jonas Vasell, CR&T, Anders Nyman, Ericsson Telecom, Ulf Ohlsson, ERD, Åke Bromö, ABB

Martin Törngren är sammankallande till styrelsens första möte.

Ordförande och kassör och ytterligare funktionärer utses vid detta möte.

§9. Till revisorer valdes Per Gunningberg, SICS och Hans Hansson, Uppsala Universitet

§10. Val av valberedning: Jörgen Hansson, LiTH och Göran Forsström, ABB valdes.

§11. Årsavgiften fastställdes till 0 kronor.
Medlemmar har möjlighet att betala 1000:- för ständigt medlemsskap.

§12. Övriga frågor:

a.Nästa seminarium 2001 - blir troligen i Halmstad.
b.Email och adresslistor - restriktiv med Emaillista, all användning genom Roland. Adress lista som förut - 500:- från kommersiella organisationer.
c. Informations spridning:
till medlemmar - epost, mera epost
SNART-konferensen - mötesplats för industri och högsola.

Sundbyholms slott 2000-08-22

Vid protokollet:

Anders Törne,
Linköpings Universitet/
Carlstedt Research & Technology

Mötesordförande:

Martin Törngren
KTH

Justeras:

Jan Jonsson
Chalmers

Karl-Erik Årzén
LTH

SNART

RTiS 2001 Topic 4

System Design: Towards System-on-Chip

The rapid development of microelectronics technology has led to the possibility of integrating an entire system on a single chip, consisting of millions of transistors. This development has, however, brought with it also enormous challenges, one of which is the need for advanced design methodology and tools to handle the complexity and heterogeneity of systems built on silicon. This session will address the challenges, the research activities, and the current industrial practice in the design of SoC for real-time applications.

The session will include the following presentations as well as a panel discussion involving all speakers.

What are the challenges for the industrial designers?

Stefan Sjöholm , ABB/MDH
This talk will address the main problems facing the industry designers and some of the current industrial practice. It will present in particular a top-down design methodology and its implication in an industrial design environment.

Mini-tutorial: Codesign of Embedded Systems: Where are we now?

Petru Eles, LiTH
Efficient embedded systems cannot be designed today without considering together both the hardware infrastructure and the software related issues. This mini-tutorial will identify the most important current trends and topics in the area of hardware/software co-design of embedded systems. It starts by shortly presenting the evolution of the area from the early 90's and then shows how the growing complexity of embedded systems and increasing requirements have changed several of the initial assumptions and influenced research directions.

The Design of a Multiprocessor SoC

Lennart Lindh, MDH
This talk will present the design and implementation of a multiprocessor SoC. It will discuss a generic platform generator as a way to reduce time-to-market and verification time.

Socware: the Swedish National System-on-Chip Initiatives

Zebo Peng , LiTH
This talk will present the Swedish national program on Socware (System-on-chip-ware), a five-year joint effort by several universities (LiTH, KTH and LTH) and institutes (Acreo) as well as industry from Sweden and abroad. The main objective of Socware is to build up competence via research and education in the design of SoC, especially for wireless and broadband communications.

Verification of Embedded Systems using a Petri Net based Representation

Luis Alejandro Cortés, Petru Eles, and Zebo Peng
Dept. of Computer and Information Science
Linköping University, Linköping, Sweden
{luico,petel,zebpe}@ida.liu.se
http://www.ida.liu.se/~luico/

Embedded systems are typically constituted of application-specific hardware components and software running on programmable platforms. The inherent heterogeneity of this kind of systems makes them very complex and consequently difficult to verify. Moreover, the increasing demand on high-performance products has boosted the levels of sophistication of such systems. For the levels of complexity typical to modern electronic systems, traditional validation techniques like simulation and testing are neither sufficient nor viable to verify their correctness. Formal methods are becoming a practical alternative to ensure the correctness of designs. They might overcome some of the limitations of traditional validation methods. At the same time, formal verification can give a better understanding of the system behavior, contributes to uncover ambiguities, and reveals new insights of the system.

The contributions of this work are two fold: first, we formally define the semantics of PRES+, a Petri net based computational model aimed to represent embedded systems; second, we introduce an approach to formal verification of embedded systems by using model checking and we propose a systematic procedure to translate PRES+ models into timed automata in order to make use of existing model checking tools.

PRES+ (Petri net based Representation for Embedded Systems) is a computational model based on Petri nets that allows to capture important features of embedded systems. When used to model embedded systems, the representation we introduce has several interesting characteristics: non-determinism may be naturally represented by PRES+; parallel or concurrent activities may be easily expressed in terms of Petri nets; in our model tokens carry information, thus PRES+ overcomes the lack of expressiveness of classical Petri nets where tokens are considered as “black dots”; our model captures timing aspects by associating lower and upper limits to the duration of activities related to transitions and keeping time information in token stamps.

Formal methods have been extensively used in software development and hardware verification. However, they are not commonplace in embedded systems design. In this work we present an approach to verification using model checking for embedded systems represented in PRES+. Model checking is an approach to formal verification that lets the designer prove whether certain design properties hold in a given model of the system. Our approach allows to determine the truth of CTL (Computation Tree Logic) and TCTL (Timed CTL) formulas with respect to a PRES+ model. In order to use existing model checking tools, we introduce a systematic procedure to translate PRES+ models into timed automata. This method can be automated in a relatively simple manner.

We study an ATM server modeled in PRES+ in order to illustrate the feasibility of our approach on practical applications. This example shows that our work is not only appropriate to verify the correctness of embedded systems, but may also be a useful tool for design space exploration.

Handling Persistent Connections in Overloaded Web Servers

Thiemo Voigt
Swedish Institute of Computer Science*
thiemo@sics.se

Per Gunningberg
Information Technology
Uppsala University
Per.Gunningberg@it.uu.se

1 Summary

Web servers have to be protected from overload since overload can lead to unpredictable response times, low throughput and even loss of service. To protect servers from overload several admission control architectures have been developed [1, 4, 5, 6, 2, 3, 8].

Many of these architectures can reject or accept connection requests based on information provided in the HTTP header [3, 8]. Persistent connections allow HTTP clients to send several requests on the same TCP connection to reduce client latency and server overhead [7]. Persistent connections are also a challenging problem with respect to admission control since the resource consumption of the requests is unknown at the time the admission control decision has to be made, i.e. when the web server receives the first request on a persistent connection. For example, a user visiting the home page of a company might either leave the company's web pages after visiting a single page or might initiate a long session. Obviously, a long session demands much more resources. Admission control is thus a trade-off. If admission policies are too restrictive, potential customers might be rejected unnecessarily resulting in loss of revenue. If one is too optimistic, the server may become overloaded with unpredictable response times and low throughput as a possible consequence. Even well-engineered adaptive overload prevention schemes suffer from this problem. For example the onset of overload may not be predicted with sufficient accuracy due to workload fluctuations. This defines the goal of our work: To avoid uncontrollable overload while maximizing access.

In [8] we presented kernel-based mechanisms that provide admission control and service differentiation based on filter rules associated with connection and application level information. We have shown that kernel-based mechanisms are more efficient and scalable than controls implemented in user space. In this paper, we extend this architecture to provide kernel-based control of persistent connections. The goal of our overload mechanism is to allow users to complete the sessions (by session, we mean a sequence of individual requests on the same TCP connection) that are regarded as important as well as sessions initiated by users that are regarded as important (for example, users having a service agreement with the site), even when the server becomes overloaded. For example, a session can be regarded as important when the client has placed some items into a shopping bag and thus the likelihood that the client will eventually purchase some items is high. On the other hand, if an unknown user is browsing the site with no evidence that the user might purchase something and an unexpected overload situation occurs, it might be more important to preserve other connections.

We have decided to judge the importance of persistent connections based on the cookies found in the HTTP header. The major advantage of using cookies is that all the information needed to determine the current importance of a connection is found in the HTTP and lower layer protocol headers. Cookie-based

*Thiemo is also at Uppsala University. This work is partially funded by the national Swedish Real-Time Systems research initiative ARTES (www.artes.uu.se), supported by the Swedish Foundation for Strategic Research.

connection control is also flexible: When the application is changed, only the filter rules associated with the affected cookies need to be updated. Furthermore, cookies do not only contain information about the current session, but also longer lasting information such as customer identification.

Based around the notion of cookie-based connection control, we extend the architecture presented in [8]. Our experiments, not shown in this summary, show that our approach can prevent server overload and also provide service differentiation under high load. Our approach can also be implemented in the web server or in a middleware layer.

References

[1] T. Abdelzaher and N. Bhatti. Web server qos management by adaptive content delivery. In *Int. Workshop on Quality of Service*, June 1999.

[2] J. Almeida, M. Dabu, A. Manikutty, and P. Cao. Providing differentiated levels of service in web content hosting. In *Proc. of Internet Server Performance Workshop*, March 1999.

[3] Nina Bhatti and Rich Friedrich. Web server support for tiered services. *IEEE Network*, September 1999.

[4] L. Cherkasova and P. Phaal. Session based admission control: a mechanism for improving the performance of an overloaded web server. Technical report, Hewlett Packard, 1999.

[5] V. Kanodia and E. Knightly. Multi-class latency-bounded web servers. In *Intl. Workshop on Quality of Service*, June 2000.

[6] K. Li and S. Jamin. A measurement-based admission controlled web server. In *Proc. of INFOCOMM*, March 2000.

[7] J. C. Mogul. The case for persistent-connection http. In *Proc. of SIGCOMM*, 1995.

[8] Thiemo Voigt, Renu Tewari, Douglas Freimuth, and Ashish Mehra. Kernel mechanisms for service differentiation in overloaded web servers. Usenix Annual Technical Conference 2001.

2 More Information

More information on the project can be found at:
http://www.sics.se/~thiemo/pamp

A more detailed HTML version of this paper can be found at:
http://www.sics.se/~thiemo/snart

Applications of Lock and Wait-Free shared data structures to Real-Time Systems

Håkan Sundell
Chalmers University of Technology
Department of Computing Science
phs@cs.chalmers.se

Abstract:

I am a member of the ARTES project called WARPing which has been running since spring 1999. My current research is within real-time operating systems with focus on shared data structures for concurrent programming and their benefits.

I will give an overview of the current research of improving current lock and wait-free algorithms, presenting the different approaches existing. I have studied how using timing information available in real-time systems can be used to improve and simplify wait-free algorithms for shared data structures. This work has resulted in two papers, presented at two major international conferences. The results form experiments and analytical evaluations seem very promising, and show significant improvements and high applicability within real-time systems. Work is currently being performed on how to further develop these ideas into general frameworks.

I am also doing studies and implementation work with known lock- and wait-free data structures. The effort will be focused on applications within real-time systems community, with the aim of incorporating for functionality inside of a commercial operating system. The possible benefits will be examined both on the user application side as well as for internal usage inside of operating system kernels.

WARPing homepage: http://www.cs.chalmers.se/~phs/warp

Task Attribute Assignment of Fixed Priority Scheduled Tasks to Reenact Off-line Schedules

Radu Dobrin and Gerhard Fohler
Department of Computer Engineering
Mälardalen University, Sweden
{radu.dobrin@., gerhard.fohler@.}mdh.se

Abstract

A number of industrial applications advocate the use of time-triggered approaches for reasons of predictability, distribution, and particular constraints such as jitter or end-to-end deadlines. The rigid offline scheduling schemes used for time-triggered systems, however, do not provide for flexibility. Fixed priority scheduling can provide more flexibility, but is limited with respect to predictability, as actual times of executions depend on run-time events.

We present a method to combine off-line schedule construction with fixed priority scheduling: by determining task attributes for the off-line scheduled tasks, such that the original schedule is reconstructed if scheduled with FPS at run-time. The method analyzes an off-line schedule together with original task constraints to create sequences and windows of tasks. Priorities and offsets are set to ensure task orders in sequences and relation between windows. As FPS cannot reconstruct all schedules with periodic tasks, our algorithm can split tasks into several instances to achieve consistent task attributes. Lower priority tasks can be added for run-time use.

1 Motivation and Contribution

Fixed priority scheduling (FPS) has been widely studied and used in a number of applications, mostly due by its simple run-time scheduling and resulting small overhead. Modifications to the basic scheme to handle semaphores [9], aperiodic tasks [10], static [11] and dynamic [7] offsets, and precedence constraints [6], have been presented. Consequently, FPS enables good flexibility for tasks with incompletely known attributes. Temporal analysis of FPS algorithms focuses on meeting deadlines, i.e., guarantees that all instances of tasks will finish *before* their deadlines. The actual times of executions of tasks, however, are generally not known and depend largely on run-time events, compromising predictability.

Off-line scheduling for time-triggered systems, on the other hand, provides strong predictability, as all times for task executions are determined and known in advance. In addition, complex constraints can be solved off-line, such as distribution, end-to-end deadlines, precedence, jitter, or instance separation. All this is enabled at the expense of loosing run-time flexibility, as all actions have to be planned before.

We present an algorithm to combine off-line schedule construction with fixed priority run-time scheduling. The resulting systems have a time-triggered base that is complemented with even-triggered on-line scheduling. This allows us to combine benefits of off-line scheduling, in particular a distributed system, complex, constrained tasks, and end-to-end deadlines, with online scheduling, which allows flexible task execution. A number of tasks are specified to execute predictable, while allowing flexibility for all others.

Our method works by transforming off-line scheduled tasks with their original constraints into tasks with attributes suited for fixed priority scheduling, i.e., periods, deadlines, and offsets, which will reenact the original offline schedule at runtime. It divides the off-line schedule and its tasks into windows and sequences, sets priorities to ensure execution orders within windows, and determines priorities and offsets to ensure orders and relations between windows. As FPS cannot reconstruct all schedules with periodic tasks, our algorithm can split tasks into several instances to achieve consistent task attributes. Tasks with lower priorities can be added for run-time scheduling.

Priority assignment for FPS tasks has been studied in, e.g., [1], [5], and [8] study the derivation of task attributes to meet a overall constraints, e.g., demanded by control performance. Instead of specific requirements, our algorithm takes an entire off-line schedule and all task requirements to determine task attributes. A method to transform off-line schedules into earliest deadline first tasks has been presented in [4].

[1] N.C. Audsley, "Optimal Priority Assignment and Feasibility of Static Priority Tasks With Arbitrary Start Times", YCS 164, Dept. Computer Science, University of York, December 1991.

[2] R. Dobrin and Y. Özdemir, “Task Attribute Assignment for Fixed Priority Scheduling”, Master’s Thesis, Department of Computer Engineering, Malardalen University, Sweden, June 2000.
[3] G. Fohler, “Flexibility in Statically Scheduled Real-Time Systems”. Ph.D. Thesis, Technisch-Naturwissen-schaftliche Fakultät, Technische Universität Wien, Wien, Österreich, April 1994.
[4] G. Fohler, “Joint scheduling of distributed complex periodic and hard aperiodic tasks in statically scheduled systems”, Proceedings of the IEEE Real-Time Systems Symposium, 1995. [5] R. Gerber, Seongsoo Hong; M. Saksena, “Guaranteeing real-time requirements with resource-based calibration of periodic processes”, IEEE Transactions on Software Engineering, Volume: 21 Issue: 7 July 1995 Page(s): 579 –592.
[6] M. Gonzalez Harbour, M.H Klein, and J.P. Lehoczky, “Fixed Priority Scheduling with Varying Execution Priority”, proceedings of the IEEE Real-Time Systems Symposium, December 1991.
[7] J.C. Palencia, M. Gonzalez Harbour, “Schedulability Analysis for Tasks with Static and Dynamic Offsets”, Proceedings of the IEEE Real-Time Systems Symposium, 1998.
[8] D. Seto, J.P. Lehoczky, Liu Sha, “Task period selection and schedulability in real-time systems” , Proceedings of the IEEE Real-Time Systems Symposium, 1998.
[9] L. Sha, R. Rajkumar, J. P. Lehoczky, ”Priority inheritance protocols: an approach to real-time synchronization”, IEEE Transactions on Computers, Volume: 39 Issue: 9, Sept. 1990 Page(s): 1175–1185.
[10] B. Sprunt, L. Sha, J. P. Lehoczky, “ Aperiodic Task Scheduling for Hard Real-Time Tasks”, The Journal of Real-Time Systems, 1989.
[11] K. Tindell, “Adding Time Offsets to Schedulability Analysis”, Technical Report YCS 221, Dept. of Computer Science, University of York, England, January 1994.

Product Line Architectures for Embedded Real-Time Systems

Anders Wall, Kristian Sandström, Jukka Mäki-Turja, Joakim Fröberg[1], and Christer Norström

Mälardalen Real-Time Research Centre, Department of Computer Engineering
Mälardalen University, Västerås, Sweden
[1]Volvo Construction Equipment AB, Sweden
{awl, ksm, jma, jfg, cen}@mdh.se

1 Introduction

Today the trend in computer-based products, such as cars and mobile phones, is shorter and shorter lifecycles. As a consequence, time spent on development of new products or new versions of a product must be reduced. One solution to this emerging problem is to *reuse* code and architectural solutions within a product family. Besides shortening development time, properly handled reuse will also improve the reliability since code is executed for longer time and in different contexts [1]. However, reuse is not trivial when applying it to real-time systems since both functional behavior as well as the temporal behavior must be considered.

We propose a design process suitable for developing product lines for real-time systems. The process starts in a requirement-capturing phase where the requirements from all products in the line are collected. Communalities in functional- and temporal requirements among the products will be considered when the actual PLA is designed. The PLA is then analyzed. The objective of analyzing the PLA is to gain confidence in that the PLA is flexible enough to be a base on which all products can be realized without violating any temporal constraints. To enable the use of a PLA and derivation of product architectures from the PLA we need a design language. Such a language is often called *architecture description language* (ADL) [2]. The ADL should have a precise syntax and semantics to enable architectural analysis, including analysis of, for instance, performance, maintainability, flexibility, and temporal properties. By making early analysis of the temporal behavior, for each product, based on the PLA and the product specific features, we can extract the following information: Traditional real-time measurements such as system utilization, response times for each feature, distribution of response times for a specific feature, and jitter information for features. However, this information is not only used for schedulability analysis, it is also used for analyzing the flexability of the PLA with respect to implementation constraints. For example, what will happen if a non-implemented component will utilize more resources than estimated in the architectural analysis?

Furthermore, robustness with respect to internal errors and erroneous assumptions about the environment can be analyzed. Typical analysis is based on "what if"-questions such as: '*What will happen if a specific component slightly overruns its time budget?*' or '*What will happen if events from the environment are generated in a higher frequency than assumed?*'

Moreover, the ADL must facilitate constructions for modeling of flexible components. The flexibility mechanisms specified on components constitutes the variation points that are used when product architectures are derived from the PLA, i.e. when the PLA is tailored for a particular product

The contributions of this work with respect to embedded real-time systems are an outline of a development process, focusing on the special considerations that must be taken into account when designing a PLA for embedded real-time systems. We also present an industrial case study of the use of a PLA for construction equipment. Finally, we propose constructions that must be supported by an ADL in order to be suitable for describing product line architectures for embedded real-time systems.

2 Product line based development

In this section we discuss the development process in which a PLA for embedded real-time products is constructed. The process is iterative and includes architectural analysis of properties that are of vital importance for a PLA, e.g. flexibility. Moreover, the derivation of products from a PLA is dealt with. The design process proposed in this paper is shown in Figure 1 where the process is divided into *requirements capturing*, *PLA development*, and *product development*. The proposed design process will be further described in this section.

Developing a PLA is done either in an *evolutionary* or *revolutionary* way [2]. The evolutionary approach to PLA design is conducted by a generalization of existing products, whereas in the revolutionary approach, the common architecture is developed, rather than extracted from existing implementations. Independent of whether the evolutionary or the revolutionary approach is taken, the first

step when developing a PLA is to capture the requirements for every product in the product line. One way to do this is to organize and group required functionality into *features*. We consider features to be functional entities seen from a stakeholder's perspective.

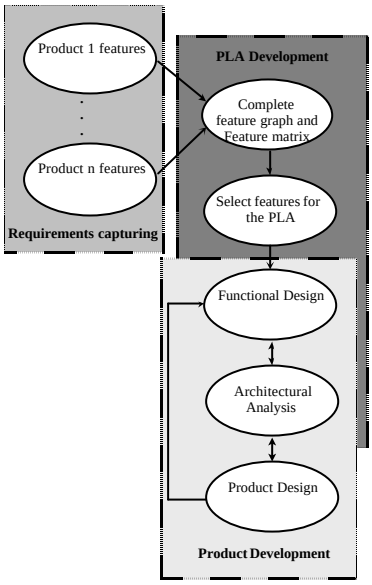


Figure 1 The process of developing a product line based on a PLA.

For instance, the function for adding items to an address book in an ordinary cellular phone can be a feature. A feature typically groups a set of requirements and therefore it also simplifies the requirements handling. At this point, we must consider all features in the product line. The products and their features make up a matrix, the *feature matrix* for the product line. The feature matrix shown in Table 1 has *m* features distributed over *n* products. A position in the matrix marked with X indicates that the feature in that column is present in the product of that line.

	Feature 1	Feature 2	Feature m
Product 1	X	X	
Product n		X	X

Table 1 A feature matrix

Features are rarely independent from each other. A feature can for instance depend on other features in order to deliver the desired functionality. Another example of a relation between features is the mutually exclusive relation, implying that only one of the related features can appear in the final product. If mutually exclusive features must co-exist in the product, effort has to be made to resolve the conflict. Features in real-time systems will also exhibit dependencies related to the temporal domain. For instance, consider the lock-free break feature and the anti-slid feature in automotive vehicles. Both features need information about the wheels' velocity, thereby having a shared temporal dependency related to the freshness of the sensor data. Such relations on features are specified in a *feature graph*.

The feature matrix and the feature graph constitute the basis for deciding what features to include in the PLA. Typically, features that are common among a majority of the products in the product line are included. Consequently, the PLA may provide features that are superfluous for some specific products. Identifying the commonality among the features for real-time systems is more complicated since we also have to take the temporal domain into consideration.

When the scope for the PLA has been decided, the features should be mapped to components that, together with their interrelations, constitute the actual architecture. This part of the development process is referred to as *functional design*. In the functional design of a PLA, the designer must take into consideration that features may have different implementations for different products. Depending on how implementations differ, the correct mechanisms for obtaining the desired flexibility must be selected.

For real-time systems we also have to consider temporal behavior. A typical example of how the temporal requirements influence the functional design is the following. Consider features that are functionally equivalent between a set of products. If these products will run on different infrastructures, i.e. operating system and hardware platform, the functionality may be partitioned among components in different ways to fulfill the timing requirements. In the high-end product we can partition a feature in such a way that it will be easy to maintain while in a low-end product we have to make an architecture that is focused on performance to be able to fulfill the timing requirements due to the limited resources in the low-end product.

After the functional design the PLA must be analyzed in order to verify that the architecture is flexible enough to facilitate all products in the product line. If not sufficiently flexible the architecture must be transformed. Thus, iterations between analysis and functional design are required. Since our focus is on real-time systems we would like to gain confidence in that the architecture is sufficiently flexible to be used in all products in the product line without violating the temporal constraints.

3 References

[1] N. E. Fenton, S. L. Pfleeger, Software Metrics: A rigorous & practical approach, International Thomson Computer Press, ISBN 0-534-95600-9, 1996

[2] L. Bass, P. Clements, R. Kazman, Software Architecture in Practice, Addison-Wesley, ISBN 0-201-19930-0, 1997

Next generation architecture for flight control systems

Kristina Ahlström *Jan Torin*
Department of Computer engineering
Chalmers University of Technology

ABSTRACT

The development of fault tolerant embedded control systems such as flight control systems, FCS, are currently highly specialized and time consuming. We introduce a conceptual architecture for the next decade control system where all control and logic is distributed to a number of computer nodes locally linked to actuators and connected via a communication network. In this way we substantially decreases the lifecycle cost of such embedded systems and acquires scalable fault tolerance.

The study is based on a FCS for JAS 39 Gripen, a multirole combat aircraft that is statically unstable at subsonic speed. All actuator nodes in our architecture are similar, hardware as well as software wise. Standardizing actuator nodes will even further decrease development and maintenance cost.

INTRODUCTION

All fault tolerance is based on redundancy. Distributed fault tolerance makes it possible to tailor the redundancy according to reliability requirements and to implement the fault tolerance at the most suitable level. Our philosophy is to cover and reconfigure permanent faults with hardware replication and to handle error detection, error processing and, transient or temporal faults with software techniques. Since redundancy is considered in a scalable software architecture, where it is possible to better utilize system resources in the distributed network, the cost for hardware redundancy is minimized. This is achieved by allocating critical software to several nodes in the network in contrast to the traditional approach, in which the same type of redundancy is added locally, i.e. by duplicating or triplicating nodes. This later approach fulfills the dependability requirements, however it is an expensive approach.

Additionally we take advantage of systems intrinsic redundancy, more or less all aircraft, combat or commercial, have redundancy in the use of primary control surfaces i.e. it is possible to maneuver and land the aircraft by commanding a reduced number of primary control surfaces. For JAS 39 Gripen this is used in the design of control laws such that one primary control surface can be failing as long as it fails in a safe way i.e. the FCS must know when and which control surface that fails and it shall fail by streamlining. (Most other embedded control systems for cars, trains etc have similar fail-safe conditions that can be used to reduce the hardware redundancy.)

DISTRIBUTED FCS

With a distributed system, there are several choices on where to put the computing power and the overall control laws and logic. Most mechanical control systems are not particularly computer processing demanding, e.g. all data processing in the flight control system of JAS 39 Gripen is handled by a single computer today. Consequently, in a distributed control system the computational capacity in the nodes

is not considered a limiting factor. Thus it is not mandatory to allocate the tasks according to processing load balance. A future FCS system should be based on distributed control in which the task allocation is optimized according a criterion of minimum bandwidth requirements on the communication system and additionally, allocated to achieve more fault tolerance and low maintenance cost.

ASSUMPTIONS

The computational capacity of microprocessors will increase over the years giving more processing power at a lower cost. Future sensor and actuator elements will be integrated either on the same silicon die or in the same package as the associated micro controller. Hence, wherever there is electronics i.e. a sensor or actuator, there is a node with processing power since the extra cost for data processing is negligible in comparison to cost of hardware devices.

The nodes are fail-silent in the temporal domain, connected through a broadcast bus, the protocol supports for membership agreement. Transient faults must be tolerated with high coverage (0.999).

SYSTEM ARCHITECTURE

Necessary nodes as well as their physical location is given by the elements connected for functionality; sensors, cockpit, nose wheel, engine, control surfaces. The critical sensor nodes and the bus are duplicated. The actuator nodes, one by each primary control surface, are simplex. This is possible since the aircraft is still controllable and able to perform safe landing with six of seven primary control surfaces. The duplicated sensors, the duplicated bus, and the simplex actuators forms the minimum hardware configuration that fulfills the two safety requirements a) less probability of critical failure than $0.5 \cdot 10^{-6}$ per flight hour and b) no single point of failure.

Sensor data available globally give the least traffic on the communication network, hence the minimum bandwidth criterion gives that tasks should be allocated to actuator nodes. We allocate critical tasks redundantly to similar actuator nodes. This gives a high degree of hardware fault detection and fault tolerance for both permanent and transient faults without cost of extra hardware. The computational overhead for doing this is negligible for the studied flight control systems.

With sensor data available globally (broadcast at the communication bus) and replicated actuator nodes with redundant calculations, the system hardware is best utilized. Additionally, the complexity as well as the development cost is reduced. Furthermore, maintainability is improved with a high degree of fault detection and the possibility of identical spare parts.

Note that the developed software of today's centralized FCS with over 10 000 flight hours can be re-used in this future distributed FCS. Hence, design fault and the discussion of N-version programs, which most certainly could be an option with seven control nodes, is not considered in our study.

The National Aerospace Program in Sweden supported this work, NFFP project no 349.

Real-Time Java Memory Management Issues

Torbjörn Ekman, Anders Nilsson, Klas Nilsson, Roger Henriksson, Anders Ive
Sven Gestegård Robertz and Anders Blomdell
Lund Institute of Technology

6 June 2001

Abstract

The Java programming language and execution environment has gained a lot of interest among developers of embedded real-time systems in the last few years. There are numerous reasons for using Java in real-time systems, such as:

- Object orientation
- Strict type checking
- Precise exception handling
- Automatic memory management
- "Write once, run anywhere"

All of which ensures safety and improves programming productivity.

Current versions of Java as provided by Sun lack real-time properties, but the National Institute of Standards and Technologies has proposed real-time extensions requirements [req99]. Based on these requirements two standard proposals have emerged, the Real-Time Specification for Java [BGB+00] and Real-Time Core Extensions [cor00]. Both target areas such as scheduling and dispatching of threads, deterministic synchronization mechanisms, safe thread termination, and asynchronous transfer of control. The proposals are not limited to the current scheduling mechanisms but are extendible with new policies.

Both proposals assume that garbage collection is not deterministic and hence not suitable for hard real-time requirements. Therefore they introduce new memory handling mechanisms with, in their opinion, better predictability. However, an investigation of the chosen strategies, and their implications convinced us that both proposals' memory management strategies should be abandoned.

1 The Real-Time Specification for Java

The specification includes a class hierarchy of seven subclasses of the abstract class MemoryArea which represents memory. One of these classes is the traditional heap, of which a singleton instance exists. Each thread is assigned a default memory area from which objects are allocated, but the programmer can explicitly allocate objects from other memory areas as well. Some of the defined memory areas are scoped memory regions with a limited lifetime. At the end of the lifetime of a scoped memory area all objects residing in that area are reclaimed. Scoped memory areas may also be nested.

Because of different lifetimes for various memory areas, strict assignment rules to maintain pointer safety are necessary. Violation of these rules causes run-time exceptions. To achieve deterministic behavior for hard real-time threads, instances of the thread class NoHeapRealTimeThread may not have reachable memory areas where objects are relocated or garbage collected. These checks and validation of assignments must be performed at runtime, although some could possibly be performed through static code analysis. The specification also includes primitives and memory areas to access physical memory addresses.

2 Real-Time Core Extensions

The specification partitions the memory in a baseline heap and core memory. Objects in core memory are real-time objects and may not access objects on the garbage collected baseline heap. Objects on the baseline heap may, with certain restric-

tions, access core objects by invoking explicitly declared Core-Baseline methods on objects residing in core memory.

Each active object is assigned a default allocation context but objects can explicitly be allocated in other contexts as well. When the lifetime of an allocation context has passed, all objects in that context are eligible for reclamation. Because there might still be references from the baseline heap to those objects, the baseline garbage collector is responsible for deleting the objects not reachable from the baseline heap. When all objects have been deleted the entire allocation context is reclaimed.

There are no limitations on which allocation contexts objects may reference. It is the responsibility of the programmer not to release an allocation context as long as there are pointers referencing it.

The specification also allows stack allocation of objects. These objects are placed on the run-time stack and are automaticly reclaimed as the scope is exited. To allocate stack objects, a set of restrictions apply and the reference must explicitly be declared stackable. Physical memory may be accessed through ports. These ports gives the programmer the possibility to directly read and modify the specified hardware address.

3 Memory Management Issues

Both proposals solve the problem with deterministic garbage collection by simply turning it off for hard real-time threads, leaving the memory management completely to the developer. They also introduce various restrictions for objects residing in different memory regions. Allowing direct access to physical memory limits the possibility to execute the same code both embedded and in simulated environments.

By introducing new memory types, and delegating the decision on which memory type to use to the programmer, there will always be a risk of introducing bugs related to manual memory management. So, while still maintaining the safety of Java, there are a number of drawbacks:

- Handling different memory areas complicates programming.
- Exceptions are likely to occur due to programming errors.
- The same program cannot be run in both embedded and simulated/virtual systems, which is a big disadvantage for, for instance, automation software.

These difficulties all stem from the assumption that real-time garbage collection cannot and should not (due to compatibility with existing kernels and threads) be used. In the proposed specifications we are then still left with much of the complexity of memory management in real-time programming that we wanted to get rid of, although in a different form.

Within our research, it has been demonstrated that predictable garbage collection, even in hard real-time, can be accomplished by the combination of incremental algorithms and scheduling of the collection work [Hen98]. The imposed overhead for automatic memory management does not affect the response times for high priority real-time threads nor allocation costs.

We currently have three projects exploring and refining different strategies for real-time garbage collection in embedded Java environments. Inspired by preliminary results, we claim that the current specification efforts concerning real-time extensions to Java are on the wrong track.

References

[BGB+00] Greg Bollella, James Gosling, Benjamin M. Brosgol, Peter Dibble, Steve Furr, David Hardin, and Mark Turnbull. *The Real-time Specification for Java*. Addison-Wesley, 2000.

[cor00] Real-Time Core Extensions. International J Consortium Specification, J Consortium, 2000.

[Hen98] Roger Henriksson. *Scheduling Garbage Collection in Embedded Systems*. PhD thesis, Department of Computer Science, Lund Institute of Technology, 1998.

[req99] Requirements for Real-time Extensions for the Java Platform. NIST Special Publication 500-243, National Institute of Standards and Technology, 1999.

Software Development Tools for Embedded Databases in Embedded Real-Time Systems*

Aleksandra Zagorac
Jörgen Hansson

Linköping University
Department of Computer Science
Linköping, Sweden
{aleza, jorha}@ida.liu.se

Dag Nyström
Christer Norström

Mälardalen University
Department of Computer Engineering
Västerås, Sweden
{dag.nystrom, christer.norstrom }@mdh.se

Project background

In the last years the deployment of embedded and mobile computing systems has increased dramatically. This is particularly true if one analyzes the growing market for cellular phones and the introduction of computers in various traditional mechanical areas, e.g., automotive systems. At the same time the amount of data that needs to be managed is growing. Current techniques adopted for storing and manipulating data objects in embedded and real-time systems are ad hoc, since they normally manipulate data objects as internal data structures in the application software, resulting in an unnecessarily costly development process with respect to design, implementation, and verification of the system. As complexity and the amount of data is growing in embedded systems, the need for a uniform, efficient and persistent way to store data becomes increasingly important, i.e., database functionality is needed to provide support for storage and manipulation of data. There are strong reasons why embedding databases into embedded systems have significant gains: reduction of development costs due to the reuse of database systems; improvement of quality in the design of embedded systems since the database provides support for consistent and safe manipulation of data, which makes the task of the programmer simpler. Consequently, this improves the overall reliability of the system. Furthermore, embedded databases provide mechanisms that support porting of data to other embedded systems or large central databases. Existing commercial embedded database systems, e.g., Polyhedra [10], RDM and Velocis [6], Pervasive.SQL [9], and Berkeley DB [11], have different characteristics and are designed with specific applications in mind. They support different data models, e.g., relational vs object-oriented model, and operating system platforms. Moreover, they have different memory requirements and provide different types of interfaces for users to access data in the database. Application developers must carefully choose the embedded database their application requires. However, finding an appropriate embedded database that matches the application requirements can in the worst case be a quite time consuming, costly, and difficult process. What would be a preferable approach is to have more tailorable and flexible system where the application designer can get an optimized database for a specific type of application, where the database provides only necessary functionality, gives a minimum memory footprint, and is highly integrated with embedded real-time systems.

Many embedded systems are also real-time systems that put special demands on the database. These demands have opened a new need for research on real-time databases. Significant amount of research has focussed on various schemes for concurrency control, transaction scheduling, and logging and recovery. Research projects that are building real-time database platforms include DeeDS [1], ARTS-RTDB [15], BeeHive [13], and RODAIN [5]. These emphasize real-time performance. While our project also addresses real-time requirements, the emphasis on how to tailor a database system for application-specific environments in the domain of embedded real-time systems.

Problem description

Traditional database systems are hard to modify or extend with new features mainly because of their monolithic structure and the fact that adding functionality results in increased system complexity. Having database systems that would allow adding or replacing functionality in its architecture in a modular (component-based) manner would be beneficial both for vendors and users. Oracle and Informix introduce limited customization of their database servers [8, 3], by allowing components for managing non-standard data types, data cartridges and DataBlade modules, to be plugged into fully functional database systems. A different approach to componentization is Microsoft's Universal Data Access Architecture [7] that unifies data access and manipulation. In this solution, an architecture is more flexible and customization ability is improved by al-

*This work is supported by ARTES (a network for real-time and graduate education in Sweden) and is done in collaboration with Upright Database Technology AB and Volvo Construction Equipment Components AB. More information about the project can be found at http://www.ida.liu.se/labs/rtslab/projects/ARTES_EmbeddedDatabases/

lowing reuse of existing components (data providers in this case) to build custom solutions on top of them, in addition to developing new components. To the best of our knowledge, the only research project focussing on construction of configurable databases is KIDS [2], which introduces configurable database composed out of components (database subsystems), e.g., object management and transaction management. Customization of this system is improved by having reusable architecture, as well as components stored in the library. Demands on development support in such configurable system are high, and KIDS has to some extent met these demands by providing a well-defined development process, available configuration support, and optional analysis tools, which are missing in other component-based database solutions.

From a real-time point of view, none of the approaches discussed enforce real-time behavior. Issues related to embedded systems such as low-resource consumption are not addressed at all. In contrast, existing component-based embedded real-time systems preserve real-time properties [12, 4, 14]. A component in existing component-based real-time systems is usually mapped to a task, e.g. passive components [12], binary components [4], and port-based object components [14] are all mapped to a task. Therefore, analysis of real-time components in these solutions addresses the problem of managing temporal attributes at a component level by considering them as task attributes [4, 12, 14]: worst case execution time, release time, deadline, precedence constraints, and mutual exclusion. Also, it is pointed out in [12] that components used in embedded systems must have explicit memory needs and power consumption requirements. Most component-based systems, both database and embedded real-time, do not provide adequate configuration and analysis support, which is vital for real-time systems. Thus there is a gap between embedded systems, real-time systems, database systems, and component-based software engineering that needs to be bridged to provide a database that can be tailored for different classes of embedded real-time systems.

Project status and goals

The focus of our research is to provide an embedded database platform that can be tailored for different real-time and embedded applications. We anticipate the following results: (i) a component-based library that holds a set of techniques and protocols for storing and manipulating data, i.e. database functionality, and (ii) a set of tools that can be used when designing and tailoring a database component for embedded real-time systems. Our research should also give a better understanding of the specification of components to be used in real-time setting. This includes functionality provided by the component, the resource demand required by the component when executed on different platforms, and rules for specifying how components can be combined and how the overall system can be correctly verified given that each component has been verified.

References

[1] S. F. Andler, J. Hansson, J. Eriksson, J.Mellin, M. Berndtsson, and B. Eftring. Deeds towards a distributed and active real-time database system. *ACM SIGMOD Record*, Volume 25, 1996.

[2] A. Geppert, S. Scherrer, and K. R. Dittrich. KIDS: Construction of database management systems based on reuse. Technical report, Department of Computer Science, University of Zurich, 1997.

[3] Informix DataBlade technology. Informix Corporation. http://www.informix.com/datablades/, March 2001.

[4] D. Isovic, M. Lindgren, and I. Crnkovic. System development with real-time components. In *Proceedings of ECOOP Workshop - Pervasive Component-Based Systems*, France, June 2000.

[5] J. Lindström, T. Niklander, P. Porkka, and K. Raatikainen. A distributed real-time main-memory database for telecommunication. In *Proceedings of the International Workshop on Databases in Telcommunications*, number 1819 in Lecture Notes in Computer Science. Springer, 1999.

[6] MBrane Inc. RDM Database Manager. http://www.mbrane.com.

[7] Universal data access through OLE DB. OLE DB White Papers. Microsoft Corporation. http://www.microsoft.com/data/oledb/, March 2001.

[8] All your data: The Oracle extensibility architecture. Oracle Technical White Paper. Oracle Corporation, 1999.

[9] Pervasive Inc. Pervasive.SQL Database Manager. http://www.pervasive.com.

[10] Polyhedra Inc. Polyhedra database manager. http://www.polyhedra.com.

[11] S. Software. Berkely db. http://www.sleepycat.com.

[12] J. Stankovic. VEST: A toolset for constructing and analyzing component based operating systems for embedded and real-time systems. Technical report, University of Virginia, 2000.

[13] J. Stankovic, S. H. Son, and J. Liebeherr. BeeHive: Global multimedia database support for dependable, real-time applications. In *Proceedings of the Second Workshop on Active Real-Time Databases (ARTDB-97)*, number 1553 in Lecture Notes in Computer Science. Springer, 1997.

[14] D. S. Stewart. Designing software components for real-time applications. In *Proceedings of Embedded System Conference*, San Jose, CA, September 2000.

[15] Y. K. Kim et al. A database server for distributed real-time systems: Issues and experiences. Technical report, Department of Computer Science, University of Virginia, 1994.

Software Synthesis Guaranteeing Timing Constraints

Tobias Amnell Elena Fersman Paul Pettersson Wang Yi

Department of Computer Systems, Information Technology,
Uppsala University, Box 325, 751 05 Uppsala, Sweden,
Web: http://www.docs.uu.se/docs/rtmv/,
E-mail: {tobiasa,elenaf,paupet,yi}@docs.uu.se.

Abstract We present recent work of the "UPPAAL-group" at the Department of Computer Systems of Uppsala University. The work aims at developing a common framework for automatic synthesis of executable code with guaranteed timing constraints, from high level design models.

Motivation

A key facility of many commercial tools for development of embedded software is to automatically synthesize fully executable code (i.e. code generation) from design models. However, few of the existing tools are capable of producing software with *predictable* timing behavior, i.e. code which is known a priori to satisfy given timing constraints. In fact, most of the commercial tools trust the software developer to resolve the timing issues that arises when the generated code should be executed on a specific target platform, such as e.g. analysing if all tasks will meet their deadlines.

In contrast, research tools for real-time systems such as UPPAAL [4] and Kronos [2] are often dedicated to analysis and abstraction of formal high-level design description. This has proven useful for finding errors and checking correctness properties in many case studies, e.g. [5], However, it provides little or no support for producing the actual program code to be executed in the final implementation.

In this work, we combine results and ideas from model-checking, scheduling, and synchronous programming to develop a framework for the development of real-time embedded systems. The goal is to support, on one hand formal specification, validation, analysis design, and on the other code synthesis that guarantees that certain timing constraints are meet when executed on the target system.

Code Synthesis

We propose a way of synthesising code from timed automata models, which has a predictable timing behavior. Inspired by the design philosophy of synchronous languages e.g. Esterel [1], we assume that the underlying real-time operating system guarantees the *synchrony hypothesis*[1]. We extend the model of timed automata by associating with each node a task (or several tasks in the general case). A task is assumed to be an executable program with two given parameters: its worst case execution time and its deadline. An example of the resulting model, called *executable timed automata*, is shown in Figure 1.

Intuitively, a discrete transition in an executable timed automaton denotes an event releasing a task and the clock constraints (guard) on the transition specifies all the possible arrival times of the associated task. When the task is released it is inserted into the ready queue of the operating system. Note that in the simple automaton shown in Figure 1, an instance of task A could be released before the preceding instance of task P has been computed. This means that the scheduling queue may contain at least P and A. In fact, instances of all four tasks may appear in the queue at the same time.

[1] That is, the operating system calls take little time compared to the execution times and deadlines of the tasks.

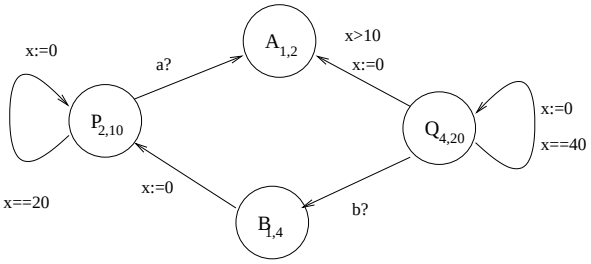


Figure 1: A system with 4 tasks, one associated with each node. The tasks P and Q are periodic with periods 20 and 40 respectively (specified by the constraints: x==20 and x==40). The tasks A and B are sporadic or event driven (by the events a and b respectively). The pair of numbers in each node give the computation times and deadlines of the task respectively.

Schedulability Analysis In the model of executable timed automata, tasks in the ready queue are executed according to a chosen scheduling strategy, e.g. earliest deadline first. In [3], it is shown that the schedulability problem can be checked by reachability analysis for non-preemptive tasks. The problem is equivalent to showing that all reachable states of the automata are schedulable. We are working on extending this result to more general execution models (e.g. preemptive tasks) and other types of analysis (e.g. maximum length of the ready queue).

Synthesizing Code for Controllers Timed automata annotated with tasks, as described above, are used as a design model of a control program. The discrete transitions (i.e the control structure) of the automaton and the associated tasks are implemented using a small set of (common) system calls and assumes that light-weight threads are provided by the underlying real-time operating system. This enables code synthesize for a variety of different target platforms. As a result, if an automaton is schedulable and the synchrony hypothesis is guaranteed by the underlying operating system, the generated code will when executed meet the constraints (timed and other) imposed on the tasks.

We have implemented a prototype that synthesizes C-code for the *legOS* operating system. *legOS* runs on the LEGO Mindstorm control brick that is equipped with an 8-bit Hitachi micro-controller. This hardware is rather typical for embedded systems of the kind we are targeting. Our prototype has so far given us promising evidence that our approach is viable.

A Small Example The implemented code generation has been used to generate a control program for th conveyor belt shown in Figure 2. A light sensor detects the color of the bricks on the belt, and a program controls the arm at the end of the belt so that all black bricks are being kicked off the belt, whereas all red bricks reaches the end of the belt.

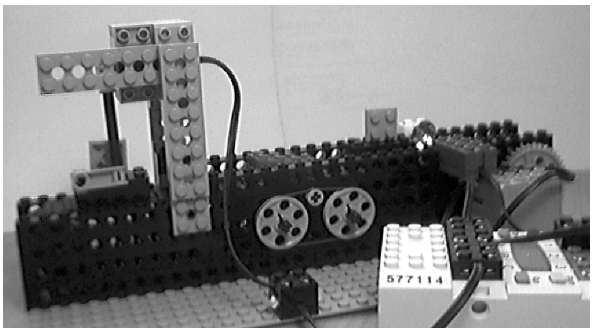


Figure 2: A conveyor belt built in LEGO.

References

[1] G. Berry and G. Gonthier. The Synchronous Programming Language ESTEREL: Design, Semantics, Implementation. *Science of Computer Programming*, 19:87–152, 1992.

[2] Marius Bozga, Conrado Daws, Oded Maler, Alfredo Olivero, Stavros Tripakis, and Sergio Yovine. Kronos: A Model-Checking Tool for Real-Time Systems. In *Proc. of the 10th Int. Conf. on Computer Aided Verification*, number 1427 in Lecture Notes in Computer Science, pages 546–550. Springer–Verlag, 1998.

[3] Christer Ericsson, Anders Wall, and Wang Yi. Timed Automata as Task Models for Event-Driven Systems. *Proceedings of RTSCA'99*, 1999.

[4] Kim G. Larsen, Paul Pettersson, and Wang Yi. UPPAAL in a Nutshell. *Int. Journal on Software Tools for Technology Transfer*, 1(1–2):134–152, October 1997.

[5] Magnus Lindahl, Paul Pettersson, and Wang Yi. Formal Design and Analysis of a Gear-Box Controller. In *Proc. of the 4th Workshop on Tools and Algorithms for the Construction and Analysis of Systems*, number 1384 in Lecture Notes in Computer Science, pages 281–297. Springer–Verlag, March 1998.

SNART

RTiS 2001 Topic 5

Communication: keeping in touch in real time!

Ulf Olsson (ERA)
One of the major changes to the way we conduct our lives - both personal and professional - has been the availability of voice communication anytime, anywhere. We are currently on the threshold of changing another fundamental aspect of modern life: not only voice, but information, as well as images (still and moving, streaming and conversational) will be equally accessible. At the same time, fixed-access bandwidth available to the general public is increasing at a tremendous rate. All this has been the subject of much futuristic optimism and, yes, hype in the previous few years. But behind and beyond that, there is a sound base of technical possibilities that we are just beginning to explore: multi-access, ad-hoc networking, quality-of-service mechanisms, the security aspects of being always connected, etcetera.

This topic is organised as follows:

Whither wireless?

Ulf Olsson , Ericsson Radio Systems AB
This talk will focus on the opportunities and the threats of merging fixed and wireless, voice/video and data all onto an IP framework, including the consequences of machine-to-machine communication becoming a major contributor to wireless usage.

What's new on the research front?

Magnus Jonsson, HH
Here, a brief walkthrough will be given outlining the major current research areas pertaining to communications technology, in particular regarding what is being done in Sweden

.

Mini-tutorial: Bluetooth

Per-Arne Wiberg; HH
In this setion, a tutorial on Bluetooth will explore the possibilities of using short-range ad-hoc networking to cover the first meter. In particular, it can be noted that Bluetooth has the potential of radically redefining the fixed vs wireless landscape.

Panel: Embedded systems on the Internet: challenges and opportunities

Finally, a panel session will be devoted to the subject of attaching embedded systems to the Internet: what are the problems and possibilities of using general-purpose IP infrastructure for automation purposes? And will the benefits outweigh the risks?

Standards for Software in Safety-related Applications

Claus Trebbien-Nielsen, DELTA
ctn@delta.dk
Lars Strandén, SP
lars.stranden@sp.se

1. Introduction

There are many standards for safety related applications in use today that specify how development of software and systems should be made in order to adhere to the requirements in specific industrial areas. However, the requirements in these standards are not co-coordinated, and differences apply for different sectors of industry. Further, since products are continuously getting more complex, more than one standard may be applicable and so the requirements of standards can be identical, complement, exclude or even contradict each other.

So a valuable insight would be to be able to evaluate a standard on its own and in relation to other standards. However it is not trivial to define a procedure for this work. In its most generic sense the comparison work is to evaluate and compare two text masses containing partly different information, with different strictness and seen from different views. The written text in itself (i.e. the implementation) contributes with complicating aspects such as:

- The text is written for humans and cannot be formalized into a version leaving no space for interpretations.
- Use of must, shall, should etc makes it difficult to isolate and compare requirements. Further, within a standard both should and shall can exist for the same item.
- Standards have different principle views e.g. if documentation is a separated process or if it is a sub process of several processes.
- Standards have different scope e.g. one can concern the contents of a product and another the way to produce it.

From above we can generally say that we do not have a single way of analyzing and comparing standards. Instead it is necessary to use both a more formal approach i.e. based on some kind of objective measure and also an informal approach i.e. some kind of evaluation and/or judgment. This will be further described below.

Since we have to limit the number of standards in this work we have selected a total of six standards used for different applications namely

- IEC 61508 "Functional safety of electrical / electronic / programmable electronic safety-related systems"
- IEC 61713 "Software dependability through the software lifecycle processes – Application guide"
- RTCA/DO-178B "Software Considerations in Airborne Systems and Equipment Certification"
- IEC 601-1-4 Medical Electrical Equipment – Part 1: General Requirements for Safety – 4. Collateral Standard: Programmable Electrical Medical Systems.
- EN 50128 Railway Applications: Software for Railway Control and Protection Systems"
- EN 954 safety of machinery: "Safety-related parts of control systems"

Also, for the work described here, SPICE (ISO/IEC TR 15504) is used as reference. Special considerations will be taken to test, verification and validation aspects.

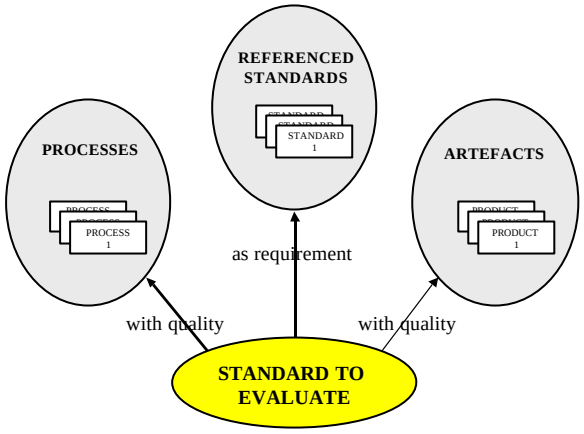
The result of this work is twofold; first, to define a pragmatic procedure that can be used for further standards as well, and second, to make a judgment of the analyzed standards and how they are related.

2. Project description

The project is ongoing and is a joint venture between DELTA Denmark and SP Sweden. The project is sponsored by NORDTEST. The final results of the survey will be published, at the latest, in December 2001. Since DELTA and SP have some different scopes this work will be approached from a process side and from an artifact side. This is of importance when describing the informal aspects of standards. If opinions differ significantly then we can conclude that a standard leaves room for interpretations and vice versa.

3. Top view

The picture below shows the top-down approach for the work within this project.



The evaluated standard may have references to other standards that are considered as requirements. These standards can in their turn be analyzed if necessary thus making a recursive procedure possible. An arrow in this case only describes if such a connection exists or not.

The evaluated standard is also mapped to processes and artifacts. The processes are initially specified by SPICE but extended if necessary. The artifacts are built up successively during analysis. Arrows in these cases include quality description i.e. informal text describing the connection. This includes for example criticality aspects, amount of connection, and contents.

Since some of the standards also define models for criticality classification these will also be discussed. Again, note that the analysis can easily be extended with more standards if necessary.

4. Single standard evaluation

The following steps are made for the analysis:

1. Map the current standard to requirements included in referenced standards.
2. Map the current standard to the set of SPICE processes. Extend it if necessary. For each connection describe the corresponding quality.
3. Map the current standard to the current set of artifacts. Extend it if necessary. For each connection describe the corresponding quality
4. Create an abstract of the standard describing its scope.

5. Standard comparison

When standards have been evaluated separately we could perform comparisons in the following ways:

- Comparing abstracts.
- Making (informal) evaluations of the standards and comparing them.
- Comparing the qualities of the connections for processes and artifacts.
- Comparing amount of referenced standards (when requirements).
- Defining metrics for the included elements within the areas (processes, artifacts and referenced standards). Using metrics for comparisons.
- Focusing the comparison on a specific aspect (here test, verification and validation).

6. Expected results

The results of the work will be recorded in a report. A number of results will be given such as:

- Description of the method for evaluation.
- Similarities between standards and completeness of each standard (i.e. how many other standards are necessary to be complete).
- Visibility of the likes and differences in formal testing requirements.
- Feasibility of SPICE framework for safety related standards.
- Applicability and comparison of models for safety criticality classification.

Or expressed more generally: to create and transfer a judgment on safety related standards to the reader with specific considerations to test, verification and validation.

Realtidslösningar för Marin Stabilisering

Presentation vid RTiS av Data Respons AB

Med marin stabilisering menas att hålla ett plan i horisontellt läge då underlaget/golvet är i rörelse som kan ha viss acceleration. Som exempel på olika applikationer kan nämnas stabilisering av kranar i offshore miljö eller anpassade, stabiliserade biljardbord för passagerar- och kryssnings fartyg.

Data Respons har i sju år utvecklat lösningar för marin stabilisering. Till att börja med mot norsk offshoreindustri. Numera har vi även ovan nämnda biljardbord.

Marin stabilisering är en hård realtidsapplikation med höga krav på operativsystem, algoritmer, sensorer och motorer. I fallet med biljardbordet är kravet att biljardbollarna skall ligga stilla, även i grov sjö. Vidare måste vibrationer minimeras och sid- och vridstabiliteten maximeras.

Vårt biljardbord som heter STable är baserat på en teknologi som är utvecklad av Data Respons och som vi kallar för ”Marin Stabiliserad Plattform” (MSP). Denna teknologi ger oss möjlighet att uppnå en nästintill accelerationsfri platform som till exempel kan ta bort accelerationer som orsakas av sidorörelser av fartyget. För att uppnå detta, så beräknas bordets rörelser som en funktion av fartygets rörelse. Mätning och beräkningar hanteras kontinuerligt av styrsystemetet. Bordets upphängning är specialltillverkad och nästan friktionsfri.

Man startar/stoppar bordet med en tryckknapp som sitter på sidan av bordet tillsammans med en liten skärm eller via en fjärrkontroll. Bordet stannar dock omedelbart om man trycker på nödstopp-knappen. Det spelar ingen roll om fartygets däck är lutande eller inte eftersom bordet alltid justerar sig automatiskt till den rätta vinkeln. Styrsystemet mäter bordets och fartygets rörelser och beräknar sedan vilka motrörelser som behövs för att hålla bordet i en stabil och nivåreglerad position. Detta är en process som måste ske kontinuerligt.

I detta projekt valde vi att använda oss av QNX (v 4.25) som realtidsoperativsystem. Eftersom vi ville använda oss av standard PC-kompatibla CPU-kort (Single Board Computers) så gav oss QNX relativt bra realtids-egenskaper samtidigt som det är ett operativsystem som har funnits länge och är stabilt. Licenskostnaden för att köra QNX är ganska liten i förhållande till produktens totala kostnad. Mjukvaran är utvecklad i C.

Applikationen på styrdatorn har tre olika användargränssnitt:

1) Ett Windows-baserat program där man genom RS232 kan ställa in och konfigurera olika parametrar i styrdatorn. Ej tillkopplad under vanlig drift

2) En liten manöverpanel som sitter monterad på sidan av bordet. Även denna panel kommunicerar med styrdatorn via RS232

3) En ”remote” manöverpanel som till exempel kan sitta bakom en bardisk eller liknande. Här använder vi oss utav RS422 eftersom det kan röra sig om längre avstånd

Från våra kunders perspektiv var det mycket viktigt att användargränsnitten är lätta att använda och robusta.

Nivåreglerningen av bordets yta verkställs av två servomotorer med tillhörande två intelligenta servoförstärkare. Hastigheten styrs från styrsystemet via en analog signal (+-5V). Däremot så gör man grundläggande inställningar av förstärkarna via ett seriellt nätverk (RS485).

Säkerhet:
Säkerheten är mycket viktigt! Största faran är att någon kommer i kläm mellan nedre kanten på sidoväggarna och golvet. Detta är löst genom att lägga klämsensorer längs med denna kant. Se bilder (svart gummi). I tillägg finns det 2 stycken nödstopp-brytare. Säkerhetssystemet är oberoende av mjukvaran och övertar styrsystemet så fort klämsensorerna eller nödstoppen aktiveras.

Hantering av I/O systemet är baserad på PC104 buss, ett egenutvecklat IO kort med viss signalbehandling i hårdvara och Arcom Elan104 CPU-kort som vi är representanter för. Data Respons har gjort flera projekt med PC 104 involverad, därför var detta en naturlig val som gäller även för hårdvara designen. Fördelen med ett sådant I/O kort är att det ger en relativ robust konstruktion på en liten yta.

Kontaktinformation:

Arne Steinarson
as@datarespons.se

Krisztian Budavari
kb@datarespons.se

Tel: 08 594 775 70 eller 08 594 775 76
Web: www.datarespons.com/stable

The OSGi platform and device access model

Subject

The presentation includes an overview of OSGi (Open Services Gateway Initiative) and Gatespace's OSGi based distributed service platform. The OSGi model for managing devices and drivers in a distributed and embedded environment will be presented. The device access model will be exemplified using 1-Wire and UPnP devices.

Contact information

Tommy Bohlin
tommy@gatespace.com
tel 0707 786400

Jan Sparud
sparud@gatespace.com
tel 0708 182143

Gatespace AB
Stora Badhusgatan 18-20
411 21 Göteborg

www.gatespace.com
tel 031 7439800
fax 031 7116416

Open Services Gateway Initiative
www.osgi.org

Abstract

OSGi defines a model for service platforms where applications written in Java can be installed and executed. The applications typically consist of a number of components - "bundles" - that together solve a problem. These bundles are provided by different vendors and must be configured to cooperate on a service platform (Framework). A large number of service platforms can be remotely administrated by an operator responsible for their continuous operation. Existing services can be upgraded and services can be installed without physical access to the platform. Great emphasis is placed on securing the platform against unauthorized access. The platform must be capable of running autonomously without requiring user interaction.

Gatespace is very active in the OSGi standardization effort and are contributing the majority of the official reference implementation. Gatespace has also developed an optimized version of the OSGi framework as well as a number of horizontal services. They include messaging, remote management, application server connectivity, a flexible user concept, media distribution, connection to payment systems and network bridging.

The base platform together with the horizontal services greatly simplify implementation of distributed services such as home entertainment applications, alarm functions, energy management and mobile services such as positioning, navigation, traffic information and weather reporting.

The platform should also mediate between applications and different types of hardware devices and provide a suitable level of abstraction of devices so that applications and devices can be independently replaced. New devices can be connected at any time (hot plugging). Suitable device drivers should automatically be activated (plug and play). A complication is that previously unknown devices may be connected, for which there are no device drivers. The OSGi device access model supports identification of new devices and additional device drivers can be downloaded on demand. This allows machines with limited local resources to support any number of device types and to always use the latest version of every driver.

The presentation gives an overview of the OSGi specification and of Gatespace's OSGi-related products and some example applications. The device access model will be demonstrated using 1-Wire and UPnP device discovery and configuration as examples.

SNART

RTiS 2001 Topic 6

Safety critical systems

This session deals with the safe operation of computers in applications where there is a threat to people, the environment, investment or goodwill. The use of safety-related computer systems is expanding rapidly into application areas including medical devices, vehicle control and monitoring, industrial process control, air traffic control and many others. The fulfillment of real-time requirements are often of vital importance for the safe operation of the systems.

In many cases, e.g. vehicle control and monitoring, there is limited previous experience of program controlled systems, especially safety critical control systems. The introduction of safety critical computer systems in new applications, and the explosion of complexity, are two major challenges facing the developers of safety-critical systems today. Important questions facing the developers which will be discussed during this session are for example; What could the new application areas learn from the areas with longer experience? What are reasonable requirements on safety? How do we know that a system is safe?

The session is divided into four parts:

1. Jan-Erik Strömberg , DST CONTROL AB
   Developing safety-critical systems, divided into
   - - an industrial perspective giving an overview of product development techniques used in a range of industrial applications, e.g. automotive electronics, medical devices, unmanned aerial vehicles, aircraft arrester systems and laser based missile defence systems
   - - a research perspective giving an overview of research trends in system development illustrated using an example aerospace application where failure mode analysis is combined with functional verification using formal methods.

   Authored by Jan-Erik Strömberg , DST CONTROL AB, Linköping, and Simin Nadjm-Tehrani , Linköping University.
2. Medical devices. This talk gives examples on requirements, methods and practices used for ensuring the safe operation of medical devices. Presented by Carl-Henry Örndal , Gambro
3. A tutorial on SIL, Safety Integrity Levels, a notion which have become very popular during recent years and which is used, although in somewhat different ways, in several recent standards. Presented by Rolf Johansson and Joakim Ohlsson , CR&T
4. A panel session discussing the challenges introduced by the introduction of safety critical computer systems in new applications, and the explosion of complexity. Within the panel session, Jan Jacobsson , SP, will contribute with experience from the application of standards and certification

---