# Licentiate Proposal
# INCENSE: Information-Centric Development of Component-Based Embedded Real-Time Systems. *

Andreas Hjertström
Mälardalen Real-Time Research Centre, Västerås, Sweden
andreas.hjertstrom@mdh.se

## Abstract

*The aim of our research is to develop new data management methods and paradigms to reduce the ever increasing complexity in today's embedded real-time systems both during development and maintenance. The research is aimed at embedded real-time systems in general and specifically towards vehicular systems. We have identified ten problem areas with data management during system development and maintenance of real-time embedded systems. As remedies for these problems we propose tools and techniques to view data as an entity on a higher level during system development and maintenance. To further improve data management, we also present a technique for how to successfully integrate real-time databases into a component-based development framework without jeopardizing basic component strategies such as reuse.*

*This work is supported by the Swedish Foundation for Strategic Research within the PROGRESS Centre for Predictable Embedded Software Systems.

## Contents

## 1. Background

The area of embedded real-time systems development are experiencing a vast increase in complexity. As an example, a high-end vehicle can contain more than 70 electrical control units (ECUs) communicating on different networks. A consequence of the increased complexity in these systems is the raising amount of signals, that can exceed 2500 [18]. This has lead to increased cost for software development to more than 23% of total manufacturing cost within the automotive industry [10]. To constrain this trend, the need for a more efficient data management is rising.

One approach to reduce complexity is Component-Based Software Engineering (CBSE). The focus within CBSE is to create software components that are reusable entities mounted together as building blocks with a possibility to maintain and improve systems by replacing individual components [6]. Vehicular industry has identified the problem of rising system complexity and developed, Automotive Open System Architecture (AUTOSAR) [12], a standard for component based software development. However, this approach does not deal with the large amount of data that needs to be managed.

The need for a more structured approach also to data management was identified in an earlier study, where they addressed this problem by developing a real-time database management system denoted COMET [21]. Real-time Databases (RTDB) [23] and RTDBMS (Real-Time Database Management System) are emerging technologies both within research society and in industry [17], aiding real-time embedded system developers to solve data management problems such as synchronization, deadlock, and persistency.

Since decreasing complexity in today's system development is such an important issue, CBSE and RTDBMS are highly interesting technologies.

## 2. Research Introduction

The purpose of this proposal is to summarize the basics concerning my research up until now as well as provide a research plan until my licentiate thesis. What has been done so far, which areas has been included in the research and what are the main results from this.

The aim of our research is to develop new data management methods and paradigms to reduce the ever increasing complexity in today's embedded real-time systems both during development and maintenance. The research is aimed at embedded real-time systems in general and specifically towards vehicular systems.

One part of the project is to join two strong upcoming techniques; real-time database management systems (RTDBMS) and component based software engineering (CBSE). To integrate an RTDBMS as a part of the component-framework is not trivial since there are several differences and conflicting aspects that differ these two. It would be desirable to achieve a component based system where data is reliably managed and structured to enable flexibility, a system where soft and hard real-time tasks can execute and keep isolation properties, a system that can handle critical transactions and at the same time enables openness for other transactions, a system where new functionality can be added or removed without side effects to the system and all communication is done through component interfaces.

One part of the project has been to solve the problem of successfully integrate RTDBMS as a part of CBSE framework without jeopardizing important properties such as, isolation and reusability. To achieve this we introduce database proxies. These proxies will work as a communication channel between the database and the components.

As stated earlier, data management in vehicle control-systems is often insufficient and lacks documentation internally in the electrical control units (ECUs) [22]. The case-study done within the scope of this project covers a broader set of companies, and focuses more on the development process and documentation of real-time data. The outcome from this case-study confirms that there is much to do within the area to achieve the satisfactory documentation, which is needed to be able to handle the needs of today and tomorrow. This is something that could be required to meet the upcoming safety regulations, eg. as specified by IEC 61508 [13], and will be a complex and difficult transition for companies in general.

Based on the outcome from the case study we will develop an information-centric design method for component-based real-time systems (design support), consisting of high level tools and design paradigms to manage and organize data in a logical view rather than a physical. During design, developers should have full control of each data item involved, who are the producers/consumers, timing requirements etc. To minimize the effort of keeping control of each data item, as much of the work as possible, connecting development and documentation should be automated. The overall aim in this part of our work is to create an information-centric design paradigm for real-time systems, where data management is treated as its own design entity.

There are a lot of research being done within the separate areas but little has been done regarding how these two strong technologies successfully can be used together.

The three main technologies used in this project is COMET RTDBMS [19], SaveCCT component technology [2] and ProSave, a component model under development based on SaveCCT.

## 3. Related Work

This section presents a short introduction to the two main techniques used in our work, Real-Time Data Base Management (RTDBMS) and Component Based Software Engineering (CBSE). Since we target embedded real-time systems and automotive systems in particular we have chosen to further investigate RTDBMS and CBSE research suitable for the domain.

### 3.1. RTDBMS

There are several different research areas within the real-time database community and there is extensive use within industry. Real-Time Database (RTDB) [23]and RTDBMS are upcoming technologies both within research society and in industry to help developers solve information management problems regarding synchronization, deadlock and persistency. Main focus in this area has mainly been towards concurrency-control, temporal consistency, overload management and scheduling.

- COMET [19] was a research project cooperation between Linköping and Mälardalen University. The focus was on real-time systems and vehicle systems in particular. COMET was intended to be used as a tightly integrated part of the control-system, providing new techniques and functionalities.

- BeeHive project [24], Real-Time and Embedded Systems Laboratory, Department of Computer Science University of Virginia is an active group within real-time databases. BeeHive's is an object-oriented database which main focus is on realtime, fault tolerance, security, and Quality of Service (QoS). Professors Sang H. Son and John A. Stankovic are the leading researchers.

- The Distributed Real-Time Systems Research Group [3], Skövde University, Sweden. They have developed a Distributed active real-time Database System (DeeDS). Main area is to study real-time database systems with soft, firm and hard deadlines, reactive mechanisms and event monitoring, as well as software timeliness testing based on techniques developed in this area.

- Department of Computer Science, University of Helsinki, have developed a distributed real-time database, RODAIN [15] (Tiina Niklander and Kimmo Raatikainen), used by the telecom industry.

- Carnegie Mellon University, Pittsburgh, developed a relational database ARTSRTDB [14] used together with a real-time operating system which supports both hard and soft real-time transactions.

- University of Texas at Dallas, Dr. Bhavani Thuraisingham [7], MITRE has done a lot of work on concurrency control algorithms using techniques such as priority ceiling and semantic locking.
- ETH Zurich Database Group [9], is together with University for Health Informatics and Technology Tyrol, working on hyperdatabases providing "higher-order data independence". This is done by guaranteeing immunity of applications against both data storage changes and in the application components and services. The future vision is for hyperdatabases to perform metadata management, scheduling, optimal routing of service requests, monitoring, flexible failure treatment, availability, and scalability.

## 3.2. CBSE

The focus within CBSE is to create software components that are reusable entities mounted together as building blocks to form systems that can be maintained and improved by replacing individual components [6].

- Autosar (AUTomotive Open System ARchitecture) [12], is a joint development project between several automotive manufactures and suppliers. The aim is to provide a common standard for software development where suppliers can integrate and maintain components throughout a products entire lifecycle.
- PECOS (PErvasive COmponent Sysyems) [11], is a component-based technology developed for reactive systems where sensors gather data and reacts by controlling actuators. PECOS aim is to provide tools and techniques for building, validate and specify software for embedded systems. A PECOS component can be either active, passive or event driven[6]. Active components control its own thread and models. A Passive component however has no thread of its own to control and are scheduled by the closest active parent. Event components are event triggered, mostly used to model hardware that periodically generates events.
- KOALA [26], is a component technology for resource constraint consumer electronics originally designed by Philips for TV products. To help developers to manage and structure configuration koala consists of an explicit architecture where it is possible with the help of an architectural description language (ADL), is a valuable tool for developers to visualize the system.
- Orion [16], A Component-Based Software Engineering Environment, a CBSE which integrates different works, involving development tools, a database repository, a process model and a middleware platform. To aid in the implementation activities, MVCASE provides wizards to automate tasks such as code generation, packaging and component publishing.
- Progress [25], Mälardalen Real-Time Center, Västerås, Sweden. Progress develops methods for engineering and re-engineering of embedded software systems. The approach is based on software-component development.
- Save [1], Mälardalen Real-Time Center, Västerås, Sweden, SAVE will develop a general framework for component-based development of safety-critical vehicular systems.
- Rubus Component Model [4], developed by Articus systems and researchers, is used today by Volvo Construction Equipment. Rubus is used for resource constrained systems and supports both soft and hard real-time embedded systems which are distinguished by two separate kernels. Hard real-time applications are time-triggered and soft real-time applications are event-triggered. Development environment is tightly integrated with Rubus operating system which makes Rubus Component Model less portable.

## 4. Problem Formulation

Component based software engineering and real-time database management systems are continuously developing separately in an effort to manage system complexity. If these two techniques are combined, the resulting system can fully benefit from the advantages of component-based software development, such as reusability, interaction through interfaces and at the same time benefit from the advantages of a real-time database management, such as openness to transactions and control systems, controlled data access, and a dynamic query language.

The aim of the thesis is to introduce an information-centric view into real-time systems development. To achieve this, new high level tools and design methods for embedded real-time systems has to be introduced where data is treated at a high level of abstraction as its own design entity. A next step will be to integrate these tools and techniques into a development framework to realize our ideas of complete information-centric development framework.

Important research questions are:

1. How can CBSE and RTDBMS be merged into a system development framework?
2. Are there sufficient tool/techniques for managing data when developing systems within industry today?
3. What tools and techniques are needed to realize an information centric design view?
4. Can efficient and flexible data management with the usage of a RTDBMS improve today's embedded system development and maintenance?
5. Can a data-centric approach where data is seen as an entity improve system development?

## 5. Project Plan

Development of embedded systems and real-time embedded systems in particular, are facing a number of difficulties with increasing data volumes, more distribution of data and a large number of computer nodes. These problem areas within data management in today's embedded real-time systems development constitute this projects main focus.

1. Thorough studies of state of the arts in relevant areas initialized our idea to combine two upcoming technologies, component based software engineering (CBSE) and real-time database management systems (RTDBMS), in an effort to aid developers and reduce system complexity. CBSE and RTDBMS has been proven to be an important step forward in embedded real-time systems development [1, 12, 20, 23, 8, 26].
2. From this research, we formulated some initial research questions and preliminary solutions. If RTDBMS can be successfully merged into the CBSE framework, benefits such as high level tools will be available to manage data. In the concept of data base proxies is introduced as a method for combining CBSE and RTDBMS without introducing any side effects.
3. To further confirm our theories we performed a case-study in cooperation with five companies was then performed to additionally validate our research direction. This study clearly shows that current practice of embedded of data management in real-time systems development is not sufficient. Our problem formulation is based on these above findings which will constitute the basis for our continued research to address and implement solutions.
4. As prof of concept we will implement or ideas into the Save demonstrator [5].

The work in this project is done within the Progress project [25] using research tools and techniques such as ProSave, SaveCCM and COMET [25, 2, 19]. The aim however is to formalize the the outcome of this research to aid embedded real-time systems development in general.

## 6. Included Papers

The licentiate thesis will be based on four papers.

- **Paper A** INCENSE: Information-Centric Run-Time Support for Component-Based Embedded Real-Time Systems, Andreas Hjertström, Dag Nyström, Mikael Åkerholm, Mikael Nolin, Proceedings of the Work-In-Progress (WIP) session, 14th IEEE Real-Time and Embedded Technology and Applications Symposium, p 4, Seattle, United States, April, 2007.

  In this paper we address our first research question and propose a technique, denoted database proxies. This technique allow the usage of real-time database management together with component-based software development without breaking fundamental rules such as reusability. A database proxy serves as a connection between components and the database. The concept of database proxies shown in Figure 1, allows components to be database unaware and therefore reusable.

  To achieve our goal and have predictable component development, flexible data access to both hard and soft real-time data, and a possibility to use powerful on-the-fly data queries we have chosen SaveCCM and Mimer SQL Real-Time Edition, (based on the COMET project) for this research.

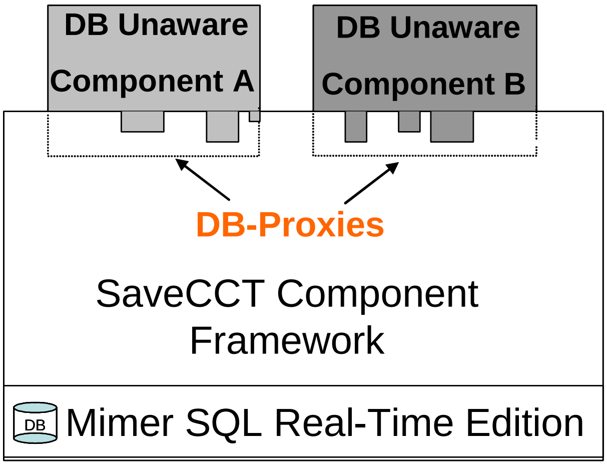


**Figure 1. Database proxies as translators between component and database**

  Status: Published.

  My contribution in this paper: I have been involved in most parts of this paper. It has been a joint effort between me and my supervisors.

- **Paper B** Design-Time Management of Run-Time Data in Industrial Embedded Real-Time Systems Development, Andreas Hjertström, Dag Nyström, Mikael Nolin.

  This paper address our research questions two and four by investigating how data is management and documented when developing and maintaining today's real-time systems. We present the results of an industrial case-study in which we have studied the state of practice in data management and documentation. Representatives from five companies within various business segments have been interviewed and our results show that various aspects of current data management and documentation are problematic and not yet mature.

  One important result from this study is that all involved companies agree that new efficient tools/techniques for documentation during development and maintenance are needed. Documentation of data internally on ECUs data is the area which has been mostly neglected.

Routines for this is inferior making adding, removing and searching for data difficult. Sometimes the whole knowledge about the internals of an ECU can reside with a single developer. Systems are becoming to complex and future rules and regulations makes it this impossible.

We have identified ten problems areas and made four main observations.

A flavor of the main identified problems:

- Lack of efficient tool support.
- Hard to verify requirements against documentation and implementation.
- Deliberately included obsolete data because of unknown dependencies.
- There are stale data in the system that no one knows exists and that could have unwanted effects to the system.
- There is a need for new techniques to facilitate development, documentation and to preserve knowledge efficient.
- Lack of visualization.

There are tools for managing distributed data but no real support for internal ECU data. A unified development environment supporting the entire development chain is needed where automated tools and techniques, physical visualization and data modeling, aid the developer.

These above areas of improvement and upcoming needs as will serve as background for my upcoming research.

Status: In submission.

My contribution in this paper: I have been the author to all parts in this paper. I have also performed the research.

- **Paper C** Data Entity Centric Development. We have already made some preliminary work on this paper and will continue to extend the information-centric view with high level tools and design paradigms, starting in September 2008. This paper will be submitted to a conference, December 2008. This paper will present ideas based on previous work on how data can be controlled at a higher level and viewed during development. Data should be seen as an entity, given a higher status during design and development. This makes it possible to specify data according to requirements already before any producer or consumer is developed. A development environment could be divided in three parts where a system architect assembles a system of components produced by a component developer and signals produced by a data administrator. In this way its not necessary for the system architect to have profound database knowledge.

  Status: Starting August 2008.

  My contribution in this paper: I intend to be the initiator to the majority of this paper and be the main author.

- **Paper D** Technical report on Comet, proxy, Save++ Demonstrator. As a result of some additional research and real implementation within the Save demonstrator project, an extension to a full paper **Paper D** based on the **paper A** and the results from the **Paper B** will be written. The implementation part of this work is already in progress and are due to finish in June 2008. The paper will include some additional information and dept on how we are going to integrate RTDBMS into a CBSE framework and successfully test it on target hardware. In this case a CCSystems CPU denoted Crossfire. We will submit this paper, either as a full paper or as a technical report in February 2009.

  Status: Ongoing.

  My contribution in this paper: I intend to be involved in most parts of the research in this paper and to be the main author.

## 7. Research Contribution

The overall aim of this project is to reduce complexity and minimize cost when developing embedded real-time systems. This involves the usage of component technology and a real-time database management system.

Our contribution in this project to embedded real-time systems development in general, is Incense, information centric development which has purpose is to reduce system complexity both on a high and low level system development. We aim to integrate tools and techniques for data management into a component framework where data is seen as an entity with its own properties.

Contributions to our research problems:

1. In paper A, we target our first and fourth research question and provide a method for integrating a RTDBMS into a component framework by using the concept of database proxies. This will allow RTDBMS to merge with CBSE and introduce valuable features such as flexibility, security and high level tools for managing data.

2. Paper B, provides valuable information to research questions two and three and clearly states that today's data management is not sufficient enough. In this paper we suggest a number of improvements and remedies to aid developers with tools and techniques.

3. Paper C, will provide answers to research questions five. In this paper we will show how development can improve by raising data managing to a higher level during development and maintenance. Data should be seen as an entity and RTDBMS as a valuable asset. There should be no difference for a system architect when developing the system if a RTDBMS or regular memory managing is used. The choice of using a RTDBMS should be a design decision.

4. In Paper D, will present the outcome of an implementation into the Save demonstrator, using the concept of database proxies. This will be performed as proof of concept to research question one and four.

## 8. Current Status

This section presents what has been achieved so far in my research.

### 8.1. Publications

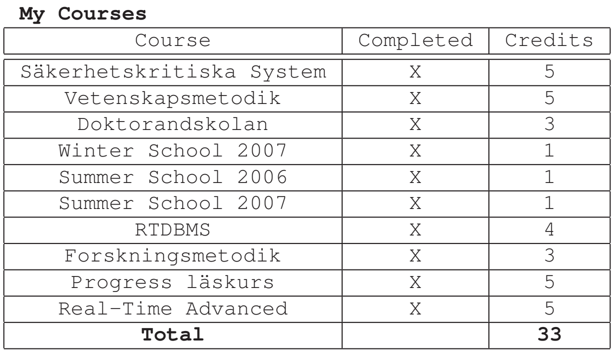
**Paper A** 2007-04-01, Incense: Information-Centric Run-Time Support for Component-Based Embedded Real-Time Systems. IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), Seattle, US.

**Paper B** (In submission). 13th IEEE International Conference on Emerging Technologies and Factory Automation, Design-Time Management of Run-Time Data in Industrial Embedded Real-Time Systems Development.

### 8.2. Additional Attended Conferences

2007-05-26, ICSE, Workshop on Software Engineering for Automotive Systems. I did not present on this highly interesting conference, but it was an important conference and an excellent opportunity to learn more about the vehicular domain and software development in general.

## 8.3. Attended Courses



**My Courses**

| Course | Completed | Credits |
|---|---|---|
| Säkerhetskritiska System | X | 5 |
| Vetenskapsmetodik | X | 5 |
| Doktorandskolan | X | 3 |
| Winter School 2007 | X | 1 |
| Summer School 2006 | X | 1 |
| Summer School 2007 | X | 1 |
| RTDBMS | X | 4 |
| Forskningsmetodik | X | 3 |
| Progress läskurs | X | 5 |
| Real-Time Advanced | X | 5 |
| **Total** | | **33** |

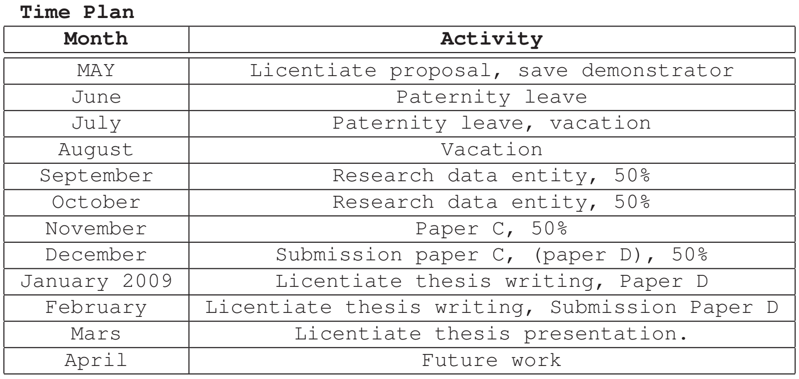
**Figure 2. An overview of courses.**

## 8.4. Planned Courses

- Statistics 5p
- Pedagogic 3p

## 8.5. Time Plan

Figure 3 gives an overview of my work path until my licentiate thesis. Some of the activities are marked 50%, this indicates a 50% paternity leave.



**Time Plan**

| **Month** | **Activity** |
|---|---|
| MAY | Licentiate proposal, save demonstrator |
| June | Paternity leave |
| July | Paternity leave, vacation |
| August | Vacation |
| September | Research data entity, 50% |
| October | Research data entity, 50% |
| November | Paper C, 50% |
| December | Submission paper C, (paper D), 50% |
| January 2009 | Licentiate thesis writing, Paper D |
| February | Licentiate thesis writing, Submission Paper D |
| Mars | Licentiate thesis presentation. |
| April | Future work |

**Figure 3. Time plan overview**

## References

[1] M. Åkerholm. A software component technology for vehicle control systems. Licentiate thesis, February 2005.

[2] M. Åkerholm, J. Carlson, J. Fredriksson, H. Hansson, J. Håkansson, A. Möller, P. Pettersson, and M. Tivoli. The Save Approach to Component-Based Development of Vehicular Systems. *Journal of Systems and Software*, 2006.

[3] 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*, 25(1):38–40, 1996.

[4] Arcticus Systems. http://www.arcticus.se.

[5] J. Carlson, J. Haakansson, and P. Pettersson. SaveCCM: An analysable component model for real-time systems. In Z. Liu and L. Barbosa, editors, *Proceedings of the 2nd Workshop on Formal Aspects of Components Software (FACS 2005)*, volume 160 of *Electronic Notes in Theoretical Computer Science*, pages 127–140. Elsevier, 2006.

[6] I. Crnkovic. Component-Based Software Engineering - New Challenges in Software Development. *Software Focus*, December 2001.

[7] Dr. Bhavani Thuraisingham, MITRE. http://www.utdallas.edu/ bxt043000/.

[8] Enea. http://www.enea.com.

[9] ETH Zurich, database research group. http://www.dbs.ethz.ch/.

[10] L. Gabriel and H. Donal. Expanding Automotive Electronic Systems. *Computer*, 35(1):88–93, Jan 2002.

[11] T. Genssler, A. Christoph, M. Winter, O. Nierstrasz, S. Ducasse, R. Wuyts, G. Arévalo, B. Schönhage, P. Müller, and C. Stich. Components for embedded software: the pecos approach. In *CASES '02: Proceedings of the 2002 international conference on Compilers, architecture, and synthesis for embedded systems*, pages 19–26, New York, NY, USA, 2002. ACM.

[12] H. Heinecke, K.-P. Schnelle, H. Fennel, J. Bortolazzi, L. Lundh, J. Leflour, J.-L. Maté, K. Nishikawa, and T. Scharnhorst. Automotive open system architecture - an industry-wide initiative to manage the complexity of emerging automotive e/e-architectures. Technical report, 2004.

[13] International Electrotechnical Commission IEC. Standard: IEC61508, Functional Safety of Electrical/Electronic Programmable Safety Related Systems. Technical report.

[14] Y.-K. Kim, M. R. Lehr, D. W. George, and S. H. Son. A Database Server for Distributed Real-Time Systems: Issues and Experiences. In *Proceedings of the Second IEEE Workshop on Parallel and Distributed Real Time Systems*, pages 66–75. IEEE Computer Society, April 1994.

[15] J. Lindstrom, T. Niklander, P. Porkka, and K. Raatikainen. A Distributed Real-Time Main-Memory Database for Telecommunication. In *Proceedings of the Workshop on Databases in Telecommunications*. Springer, 1999.

[16] D. Lucrédio, C. de Paula Bianchini, A. F. do Prado, L. C. Trevelin, and E. S. de Almeida. Orion Ű a component-based software engineering environment. *JOURNAL OF OBJECT TECHNOLOGY*, vol.3(no.4), 2004.

[17] Mimer SQL Real-Time Edition, Mimer Information Technology. Uppsala, Sweden. http://www.mimer.se.

[18] N. Navet. Trends in Automotive Communication Systems. In *Proceedings of the IEEE*, volume 93, pages 1204–1223, June 2005.

[19] D. Nyström. Comet: A component-based real-time database for automotive systems.

[20] D. Nyström. *Data Management in Vehicle Control-Systems*. PhD thesis, Mälardalen University, October 2005.

[21] D. Nyström, M. Nolin, A. Tešanović, C. Norström, and J. Hansson. Pessimistic Concurrency Control and Versioning to Support Database Pointers in Real-Time Databases. In *Proceedings of the 16th Euromicro Conference on Real-Time Systems*, pages 261–270. IEEE Computer Society, June 2004.

[22] D. Nyström, A. Tešanović, C. Norström, J. Hansson, and N.-E. Bånkestad. Data Management Issues in Vehicle Control Systems: a Case Study. In *Proceedings of the 14th Euromicro Conference on Real-Time Systems*, pages 249–256. IEEE Computer Society, June 2002.

[23] K. Ramamritham, S. H. Son, and L. C. Dipippo. Real-Time Databases and Data Services. *Journal of Real-Time Systems*, 28(2/3):179–215, November/December 2004.

[24] J. A. Stankovic, S. H. Son, and J. Liebeherr. *Real-Time Databases and Information Systems*, chapter BeeHive: Global Multimedia Database Support for Dependable, Real-Time Applications, pages 409–422. Kluwer Academic Publishers, 1997.

[25] The PROGRESS Centre for Predictable Embedded Software Systems. http://www.mrtc.mdh.se/progress/.

[26] R. van Ommering. *Building Reliable Component-Based Software Systems*, chapter The Koala Component Model, pages 223–236. Number ISBN 1-58053-327-2. Artech House Publishers, July 2002.