Mälardalen University Press Licentiate Theses
No. 146

MODELING AND TIMING ANALYSIS OF INDUSTRIAL COMPONENT-BASED DISTRIBUTED REAL-TIME EMBEDDED SYSTEMS

Saad Mubeen

2012

MÄLARDALEN UNIVERSITY
SWEDEN

School of Innovation, Design and Engineering

ISBN 978-91-7485-055-0
ISSN 1651-9256
Printed by Mälardalen University, Västerås, Sweden

Abstract

The model- and component-based development approach has emerged as an attractive option for the development of Distributed Real-time Embedded (DRE) systems. Within this context we target issues related to modeling of legacy communication, extraction of end-to-end timing models and support for holistic response-time analysis of industrial DRE control systems.

We introduce a new approach for modeling legacy network communication in component-based DRE systems. By introducing special-purpose components to encapsulate and abstract the communication protocols in DRE systems, we allow the use of legacy nodes and legacy protocols in a component- and model-based software engineering environment. The proposed approach also supports the state-of-the-practice development of component-based DRE systems. Because an end-to-end timing model should be available to perform the holistic response-time analysis, we present a method to extract the end-to-end timing models from component-based DRE systems.

The Controller Area Network (CAN) is one of the widely used real-time networks in DRE systems especially in automotive domain. We identify that the existing analysis of CAN does not support common message transmission patterns which are implemented by some high-level protocols used in the industry. Consequently, we extend the existing analysis to facilitate the worst-case response-time calculation of these transmission patterns. The extended analysis is generally applicable to any high-level protocol for CAN that uses periodic, sporadic, and both periodic and sporadic transmission of messages.

In order to show the applicability of our modeling techniques and extended analysis, we provide a proof of concept by extending the existing industrial component model (Rubus Component Model), implementing the holistic response-time analysis along with the extended analysis of CAN in the industrial tool suite (Rubus-ICE), and conducting an automotive-application case study.

To my mother

Acknowledgements

First of all, I would like to express my deepest gratitude to my supervisors Professor Mikael Sjödin and Dr. Jukka Mäki-Turja. The work presented in this thesis would not have been possible without their expert guidance, persistent help and tremendous encouragement. I am grateful to them for providing valuable and useful suggestions for improvement of this thesis. I had a great opportunity of learning so many new things from them during our meetings and discussions.

Many thanks to the people from industry who were involved in the work presented in this thesis. Thank you Kurt-Lennart Lundbäck, John Lundbäck, Staffan Sandberg and Jimmy Westerlund.

I would like to thank Dr. Jan Carlson for co-authoring a paper and providing me useful feedback on my thesis proposal. I also thank Farhang Nemati for providing me useful tips on the structure of my thesis.

I attended several courses during my Licentiate studies. I thank Hans Hansson, Thomas Nolte, Emma Nehrenheim, Mikael Sjödin, Jukka Mäki-Turja, Ivica Crnkovic, Jan Torin, Sasikumar Punnekkat, and Kristina Lundqvist for their guidance during my studies. I want to also thank other faculty members Paul Pettersson, Jan Gustafsson, Björn Lisper, Mats Björkman, Jan Carlson, Damir Isovic, Dag Nyström, Cristina Seceleanu, Gordana Dodig-Crnkovic, Mikael Ekström, Andreas Ermedahl. You all have been a source of inspiration for me.

I would also like to thank my friends and colleagues at the department for all the fun we had during my studies, conference trips, coffee breaks and parties. I wish to thank Abhilash, Adam, Adnan, Aida, Amine,Ana, Andreas G., Andreas H., Andreas J., Aneta, Antonio, Barbara, Batu, Bob (Stefan), Danial, Eduard, Etienne, Farhang, Federico, Frank, Giacomo, Hang, Huseyin, Jagadish, Johan, Josip, Juraj, Jörgen, Lars, Leo, Luis (Yue), Luka, Mehrdad, Mikael Å, Mobyen, Moris, Nikola, Nima, Ning, Radu, Rafia, Raluca, Sara D., Severine, Shahina, Stefan B., Svetlana, Thomas L., Tibi, and others for all the fun and memories.

I also thank all the administrative staff, in particular Gunnar Widforss, Malin Rosqvist, Åsa Lundkvist, Carola Ryttersson, Sussane Fronnå for making many things easier.

Last but not least, I would like to thank my family. I thank my parents for their endless love, support and encouragement throughout my life. I am thankful to my wife for her care, support and cooperation.

This work has been supported by the Swedish Knowledge Foundation (KKS) within the project EEMDEF and the Swedish Foundation for Strategic Research (SSF) with the centre PROGRESS. I would like to thank the industrial partners Arcticus Systems and BAE Systems Hägglunds.

Saad Mubeen
Västerås, January, 2012

List of Publications

Papers Included in the Licentiate Thesis[1]

Paper A *Analyzable Modeling of Legacy Communication in Component Based Distributed Embedded Systems*. Saad Mubeen, Jukka Mäki-Turja, Mikael Sjödin and Jan Carlson. In proceedings of the 37th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), pages 229-238, Oulu, Finland, September, 2011.

Paper B *Extraction of End-to-end Timing Model from Component- Based Distributed Real-Time Embedded Systems*. Saad Mubeen, Jukka Mäki-Turja and Mikael Sjödin. In proceedings of the International Workshop on Time Analysis and Model-Based Design, from Functional Models to Distributed Deployments (TiMoBD) located at Embedded Systems Week, Taipei, Taiwan, October, 2011.

Paper C *Extending Schedulability Analysis of Controller Area Network (CAN) for Mixed (Periodic/Sporadic) Messages*. Saad Mubeen, Jukka Mäki-Turja and Mikael Sjödin. In proceedings of the 16th IEEE Conference on Emerging Technologies and Factory Automation (ETFA), pages 1-10, Toulouse, France, September, 2011.

Paper D *Support for Holistic Response-time Analysis in an Industrial Tool Suite: Implementation Issues, Experiences and a Case Study*. Saad Mubeen, Jukka Mäki-Turja and Mikael Sjödin. Accepted for publication in proceedings of the 19th IEEE Conference on Engineering of Computer Based Systems (ECBS), Novi Sad, Serbia, April, 2012.

[1] The included articles have been reformatted to comply with the licentiate layout

Additional Papers, Not Included in the Licentiate Thesis

Journals

- *Introducing Components for Modeling Real-Time Network Communication in the Rubus Component Model.* Saad Mubeen, Jukka Mäki-Turja and Mikael Sjödin. Accepted for publication in the Information Journal, International Information Institute, March, 2012.

- *Tracing Event Chains for Holistic Response-Time Analysis of Component Based Distributed Real-Time Systems.* Saad Mubeen, Jukka Mäki-Turja and Mikael Sjödin. In the ACM SIGBED Review, vol. 8, issue 3, pages 48-51, ACM, September, 2011.

Conferences

- *Response-Time Analysis of Mixed Messages in Controller Area Network with Priority- and FIFO-Queued Nodes.* Saad Mubeen, Jukka Mäki-Turja and Mikael Sjödin. In submission.

- *Towards Modeling and Holistic Timing Analysis of Industrial Component Based DRE Systems.* Saad Mubeen, Jukka Mäki-Turja and Mikael Sjödin. Accepted for publication in proceedings of the 19^{th} IEEE Conference on Engineering of Computer Based Systems (ECBS), Novi Sad, Serbia, April, 2012.

- *Implementation of Holistic Response-Time Analysis in Rubus-ICE: Preliminary Findings, Issues and Experiences.* Saad Mubeen, Jukka Mäki-Turja and Mikael Sjödin. In proceedings of the 32^{nd} IEEE Real-Time Systems Symposium (RTSS), WIP, pages 9-12, Vienna, Austria, December, 2011.

- *Extending Response-Time Analysis of Controller Area Network (CAN) with FIFO Queues for Mixed Messages.* Saad Mubeen, Jukka Mäki-Turja and Mikael Sjödin. In proceedings of the 16^{th} IEEE Conference on Emerging Technologies and Factory Automation (ETFA), pages 1-4, Toulouse, France, September, 2011.

- *Exploring Options for Modeling of Real-Time Network Communication in an Industrial Component Model for Distributed Embedded Systems*. Saad Mubeen, Jukka Mäki-Turja and Mikael Sjödin. In the Lecture Notes in Electrical Engineering (LNEE), Vol. 102, Springer, pages 441-458, August, 2011.

- *Designing Efficient Source Routing for Mesh Topology Network on Chip Platforms*. Saad Mubeen and Shashi Kumar. In proceedings of the 13^{th} Euromicro Conference on Digital System Design, Architectures, Methods and Tools (DSD), pages 181-188, Lille, France, September, 2010.

- *High Precision Response Time Analysis of Tasks with Precedence Chains*. Saad Mubeen, Jukka Mäki-Turja and Mikael Sjödin. In proceedings of the 22^{nd} Euromicro Conference on Real-Time Systems (ECRTS), WIP, pages 21-24, Brussels, Belgium, July, 2010.

Workshop

- *Modeling of Legacy Communication in Distributed Embedded Systems*. Saad Mubeen, Jukka Mäki-Turja and Mikael Sjödin. In proceedings of the 2^{nd} Workshop on Model Based Engineering for Embedded Systems Design (M-BED), located at Design, Automation & Test in Europe (DATE) Conference, pages 1-6, Grenoble, France, March, 2011.

MRTC reports

- *Implementation of Holistic Response-time Analysis in Rubus-ICE*. Saad Mubeen, Jukka Mäki-Turja and Mikael Sjödin. Technical Report ISSN 1404-3041 ISRN MDH-MRTC-258/2012-1-SE, Mälardalen University, Sweden, January, 2012.

- *Response-Time Analysis of Mixed-Type Controller Area Network (CAN) Messages*. Saad Mubeen, Jukka Mäki-Turja and Mikael Sjödin. Technical Report ISSN 1404-3041 ISRN MDH-MRTC-259/2012-1-SE, Mälardalen University, Sweden, January, 2012.

Contents

I

Thesis

Chapter 1

Introduction

In this thesis we introduce a new approach for modeling legacy network communication in component-based Distributed Real-time Embedded (DRE) systems. By introducing special-purpose components to encapsulate and abstract the communication protocols in DRE systems, we allow the use of legacy nodes and legacy protocols in a component- and model-based software engineering environment. The proposed approach also supports the state-of-the-practice development of component-based DRE systems. Because an end-to-end timing model should be available to perform the holistic response-time analysis, we also provide a method to extract such models from component-based DRE systems.

1.1 Background

An embedded system is a microprocessor-based system that is designed to perform a dedicated functionality by means of hardware and software [1]. Often, embedded systems interact with their environment through sensors and actuators. They mostly remain hidden in their applications, for example, an embedded system in a vending machine, because they are embedded inside the larger system which they control or which they are part of. They are found in almost all electronic items ranging from simple consumer products such as microwave oven and coffee machine to highly sophisticated systems such as industrial process controllers and smart phones. Their applications span over many domains such as automotive, aerospace, consumer electronics, biomedical, military, business, industrial control, and many more.

It is estimated that about 10 billion processors are manufactured every year. Out of which, approximately 99% are embedded processors while only 1% find their way to the general-purpose computers such as PCs and laptops [1, 2]. Not only the number of embedded processors has increased in the past few years, but also the software which runs on them. The embedded software has drastically increased in size and complexity. In automotive domain, for example, a modern premium car contains nearly 100 million lines of code that run on about 70 to 100 embedded processors [3]. Another example of the complexity and large size of embedded software can be seen in the software for radio and navigation system in a modern premium car such as Mercedes Benz S-Class that alone contains 20 million lines of code [3]. Because of this trend of continuously increasing size and complexity of embedded software, the development of embedded systems has become very complex.

Often, an embedded system needs to interact with its environment in a timely manner, i.e., the embedded system is a real-time system. For such a system, the desired and correct output is one which is logically correct as well as delivered within a specified time (e.g., a deadline). One way to classify a real-time system is as being either soft or hard. In soft real-time systems, infrequent deadline misses can be tolerated. For example, electronic window control system in a car is a soft real-time system. On the other hand, missing a deadline in a hard real-time system can result in the system failure. In hard real-time systems, a logically correct but late response is considered as bad as logically incorrect response. The electronic engine control system in a car is an example of a hard real-time system. Many hard real-time systems are also safety critical which means that the system failure can result in catastrophic consequences such as endangering human life or the environment. For example, airbag control system in a car is a safety-critical hard real-time system.

In order to capture, e.g., requirements early during the development, handle the complexity of embedded software, lower the development cost, reduce the time-to-market and time-to-test, allow reusability, and support modeling at higher level of abstraction, the research community proposed model- and component-based development of embedded systems by employing the principles of Model-Based software Engineering (MBE) and Component-Based Software Engineering (CBSE) [4, 5]. MBE provides the means to use models throughout the process of system development. It uses models to describe functions, structures and other design artifacts. Whereas, CBSE facilitates the development of large software systems by integration of software components. CBSE raises the level of abstraction for software development and aims to reuse software components and their architectures. There is a great interest for bringing these development techniques in the embedded systems industry [5, 6].

In DRE systems, the functionality is distributed over many nodes (processors). The nodes in a DRE system are connected to one or more networks. The software development of DRE systems is much more complex compared to uniprocessor embedded real-time systems because of various reasons including the distribution of functionality and real-time requirements on network communications. The example of a modern premium car, that we discussed above, provides a good example of an application of DRE systems. The size of embedded software in a modern premium car may reach up to 1 GB which may be realized by more than 2000 software functions distributed over 70 to 100 Electronic Control Units (ECUs) that may be connected by more than five different buses (or networks) [7].

When MBE and CBSE are used for the development of DRE systems, modeling of communication infrastructure arises as a challenge. In the industry, DRE systems are built often using legacy (sub) systems (i.e., previously developed) which use predefined rules for communication. Furthermore, DRE systems are often expected to use legacy network protocols for real-time communication. A component technology for the development of DRE systems should abstract the application software from the communication infrastructure. Moreover, the component technology should support the modeling and analysis of legacy communications and legacy systems.

The safety-critical nature of many DRE systems requires evidence that the actions by the system will be provided in a timely manner, i.e., each action will be taken at a time that is appropriate to the environment of the system. Therefore, it is important to make accurate predictions of the timing behavior of such systems. In order to provide evidence that each action in the system will meet its deadline, *a priori* analysis techniques such as schedulability analysis have been developed by the research community. The Holistic Response-Time Analysis (HRTA) [8] is a schedulability analysis technique which calculates upper bounds on the response times of event chains that are distributed over more than one node in a DRE system. The end-to-end timing model of a DRE system should be available to perform HRTA. Ideally, a component technology for the development of DRE systems should support automatic extraction of such timing model.

There are a number of real-time network protocols used in DRE systems. Among them, Controller Area Network (CAN) [9] is one of the most frequently used especially in automotive domain. It has been standardized by the International Organization for Standardization as ISO 11898-1 [10]. According to CAN in Automation (CiA) [11], the number of CAN enabled controllers sold in 2011 are estimated to be 850 million. In total, more than two billion CAN controllers have been sold until today. Out of this huge number, approximately 80% CAN controllers have been used in automotive domain. CAN is a multi-master, event-triggered, serial communication bus protocol supporting bus speeds of up to 1 mega bits per second. In this thesis, we will focus only on CAN and some of its high-level protocols which are developed for various industrial applications. These include CAN Application Layer (CAL) [12], CANopen [13], Hägglunds Controller Area Network (HCAN) [14], CAN for Military Land Systems domain (MilCAN) [15], etc.

1.2 Problem Statement and Research Questions

The model- and component-based development has emerged as an attractive option for the development of software for DRE systems. The majority of existing model- and component-based development approaches allow for structural and functional modeling. They do not support execution modeling which is concerned with the modeling of run-time properties and/or requirements (e.g., end-to-end deadlines, jitter, etc.) of software functions. The modeling of DRE systems should extend down to the execution level to allow precise control of resource utilization and that timing requirements are not violated when the system is executed. However, providing such modeling support for DRE systems is very challenging because the functionality in DRE systems can be realized with more than one execution model, e.g., separate execution models for the nodes and networks. Today, one of the main focus points during the development of DRE systems in the industry is to model and express timing related information and perform timing analysis [16].

One way to deal with these challenges is to use a component technology that allows model- and component-based development of DRE systems with the support for modeling, analyzing, predicting and modifying the execution behavior. Such a component technology should complement structural and functional modeling with the modeling of execution requirements at an abstraction level close to the functional specification while abstracting the implementation details. The component technology should allow the expression of timing related information during the development. Moreover, it should facilitate the identification of timing errors early during the development by easily rendering the modeled DRE applications for end-to-end timing analysis.

However, building such a component technology to support the state-of-the-practice development of DRE systems raises many challenges. One of the main reasons behind these challenges is that the development process of DRE systems in academia and industry may be very different from each other. In academia, the development process often starts with discussions about models and functions. The models are assumed to be platform independent. Further, it is assumed that the models and functions will be deployed on specific platforms at a later stage. However, this way of development for DRE systems is often not practiced in the industry, especially in automotive or vehicle domain. The traditional process for the development of DRE systems in the industry starts with designing the bus (or network) communication. The infrastructure for the DRE system to be developed is already known. In the early stage of industrial development process of DRE systems, usually the focus is on finding the answers to the questions as follows. How many busses will be there in the system? Which nodes will be connected to which bus? How many messages will be there in the system? Which messages will be sent by each node? After finding the answers to these questions, the focus is shifted towards the development of functions. Thus, a communication-oriented development process is used for DRE systems and constitutes the state of the practice.

In order to provide a model- and component-based approach to support the state-of-the-practice development of DRE systems, we will target the challenges concerned with the modeling of real-time network communication and support for holistic timing analysis. One such challenge is to support the modeling of legacy network communication and allow the use of legacy nodes in component-based DRE systems. In order to ensure that the DRE system will behave in a timely manner during its execution, we need to analyze tasks, messages and event chains in distributed transactions and predict the end-to-end delays. The component technology for the industrial development of DRE systems should support state-of-the-art real-time analysis such as Holistic Response-Time Analysis (HRTA). The supported HRTA should be able to incorporate the analysis of common message transmission patterns that are implemented by the real-time network protocols used in the industry. In order to perform HRTA, the end-to-end timing model of DRE systems should be available. The extraction of end-to-end timing model from component-based DRE systems is another challenge that we will target.

The research problem addressed in this thesis can be formulated as follows.

> *Investigate how to provide a model- and component-based approach for communications-oriented development of DRE systems with a support for legacy communication protocols, legacy nodes and holistic response-time analysis.*

We further refine this problem to formulate two questions that we will investigate in this thesis.

1. How to model legacy network communication and allow the use of legacy nodes for the state-of-the-practice development processes for component-based DRE systems?

2. How to extract end-to-end timing models from component-based DRE systems that are built using the state-of-the-practice development processes?

1.3 Thesis Outline

The thesis is organized into two parts:
Part I includes first three chapters. In Chapter 1 we provided an introduction to the thesis and formulated the research problem. In Chapter 2 we discuss the contributions in the thesis. Chapter 3 presents the conclusion and suggestions for the future work.
Part II presents the technical contributions of the thesis in the form of four papers which are organized in Chapters 4-7.

Chapter 2

Technical Contributions

This thesis presents the development and implementation of new modeling and timing analysis techniques which can be used for the state-of-the-practice development of component-based DRE systems. The contributions in this thesis are organized in four parts. In the first part, we introduce a new technique for modeling legacy network communication in DRE systems. The detailed contribution in this part is discussed in Paper A (Chapter 4). In the second part, we present a method to extract the end-to-end timing models from component-based DRE systems. The detailed contribution in this part is discussed in Paper B (Chapter 5). In the third part, we identify a need for the extension of existing response-time analysis of CAN, and accordingly, we present the extended analysis. The detailed contribution in this part is discussed in Paper C (Chapter 6). Finally, in the fourth part, we provide a proof-of-concept implementation of the techniques developed in previous three parts. The detailed contribution in the fourth part is discussed in Paper D (Chapter 7). In this chapter we provide a summary of these contributions.

Personal Contribution. The research work presented in these contributions was done in collaboration with my supervisors Prof. Mikael Sjödin and Dr. Jukka Mäki-Turja along with Dr. Jan Carlson (only Paper A). I am the main contributor and first author of all the papers.

2.1 Modeling of Legacy Network Communication in Component-based DRE Systems

This contribution addresses first research question. We introduce a new approach for modeling real-time network and legacy communication in component-based DRE systems. In order to show usability of our modeling approach, we implement it by extending the existing industrial component model, i.e., Rubus Component Model (RCM) [17]. By introducing special-purpose components to encapsulate and abstract the communication protocols in DRE systems, we allow the use of legacy nodes and legacy protocols in a component- and model-based software engineering environment. With the addition of these components, RCM will be able to not only model real-time network communication, but also support state-of-the-practice development of component-based DRE systems. The proposed extension also allows model- and component-based development of new nodes that are deployed in legacy systems that use predefined communication rules. These extensions also enable adaptation of a node when communication rules change (e.g., due to re-deployment in a new system or due to upgrades in the communication system) without affecting its internal component design. The special-purpose components can be automatically generated from the information about legacy communication or from early design decisions about network communication. Although RCM was selected for the proof-of-concept implementation, the proposed extensions should be generally applicable for the extension of several component models for the development of DRE systems that use the pipe-and-filter style for component interconnection such as ProCom [18] and COMDES-II [19].

2.2 Extraction of End-to-end Timing Models

This contribution addresses second research question. HRTA is an important activity during the development of DRE systems. In order to perform HRTA of component-based DRE systems, the end-to-end timing models should be extracted from them. The extraction of such models can be challenging because the design and analysis models are usually built using different meta-models. We present a method to extract the end-to-end timing models from component-based DRE systems to facilitate HRTA. This method is built upon the modeling approach that we discussed in the first contribution (Paper A). We discuss and solve the issues concerning the model extraction such as extraction of unambiguous timing and tracing information from all nodes and networks in the system and tracing of event chains in distributed transactions. The extraction method for end-to-end timing models and the solutions of encountered problems may be applied to several component models that use a pipe-and-filter style for component interconnection. The end-to-end timing model that we considered is also general as it incorporates the analysis of several real-time network protocols used in the automotive domain. To show the applicability of our approach, we demonstrate the implementation of end-to-end timing model extraction in the analysis framework of the existing industrial tool suite Rubus-ICE [20].

2.3 Extension of the Existing Analysis for Controller Area Network

To analyze communications in DRE systems, it is important to find out whether the existing analysis is sufficient or extensions are required to meet the industrial needs. In this work, we focus only on CAN and some of its high-level protocols. While answering the two research question (discussed in Chapter 1), we identified that the existing response-time analysis of CAN does not support the analysis of common message transmission patterns which are implemented by some high-level protocols used in the industry. The existing analysis calculates the response times of CAN messages that are queued for transmission periodically or sporadically. However, there are a few high-level protocols for CAN such as CANopen and HCAN that support the transmission of mixed messages as well. A mixed message can be queued for transmission both periodically and sporadically. In other words, a mixed message is simultaneously time and event triggered. Thus, it may not exhibit a periodic activation pattern. In order to support the development of DRE systems employing high-level protocols for CAN, there is a need to extend the existing analysis. We extend the existing response-time analysis of CAN to support mixed messages. The extended analysis is generally applicable to any high-level protocol for CAN that uses periodic, sporadic, and both periodic and sporadic transmission of messages.

2.4 Proof-of-Concept Implementation

In this contribution we validate our solutions to the research questions. In order to transfer the new modeling techniques and extended analysis, discussed in the previous three contributions, to the industry we need to validate them first.

While validating our solutions, we found out that the process of implementing and integrating state-of-the-art real-time analysis with an existing industrial tool suite offers many challenges. The Implementer has to not only code and implement the analysis in the tool suite, but also deal with several other issues. We present the implementation of HRTA as a plug-in for the existing industrial tool suite Rubus-ICE. As part of HRTA, we implemented the existing as well as the extended analysis discussed in the third contribution. The implemented HRTA is general as it supports the integration of response-time analysis of various networks without a need for changing the holistic algorithm. We discuss and solve encountered issues and highlight gained experiences during the implementation, integration and evaluation of HRTA plug-in. We believe that most of the experiences gained and solutions to the issues encountered in this work maybe applicable when other complex real-time analysis techniques are implemented in any industrial tool suite that supports a plug-in framework (for the integration of new tools) and component-based development of DRE systems. Finally, we provide a proof of concept for all modeling approaches and extended analysis discussed in this thesis by modeling an automotive industrial application (autonomous cruise control system) using extended RCM and analyzing it with HRTA plug-in in Rubus-ICE.

2.5 Discussion

We selected RCM and accompanying tool suite Rubus-ICE to provide a proof-of-concept implementation of our new modeling techniques and extended analysis for several reasons. Among them, one reason is the existing support for structural, functional and execution modeling of dependable embedded real-time systems. Further, RCM and Rubus-ICE provide a means for developing predictable and analyzable control functions with a support for modeling real-time properties and requirements, interconnections between the functions in terms of data flow and control flow separately, and generation of run-time framework.

With the proposed extensions, RCM along with Rubus-ICE can be considered a suitable choice for the component-based development of DRE systems in the industry for many reasons. For example, it complements the structural and functional modeling with the execution modeling of DRE systems; it supports communications-oriented development process for DRE systems; it supports the modeling of legacy communication and legacy systems; it can easily model and specify the timing related information; it has a small run-time footprint (timing and memory overhead); it implements the state-of-the-art research results; and it has a strong support for development and analysis tools.

2.6 Impact of Contributions

The new approaches for modeling legacy network communication and extraction of end-to-end timing models may be suitable for other component models, for DRE systems, that use a pipe-and-filter style for component interconnection. The extended analysis supports common message transmission patterns that are implemented by several high-level protocols used in the industry today. Further, the analysis engines support the integration of the analysis of various real-time networks without a need for changing the holistic algorithm. Most of the encountered issues, proposed solutions and gained experiences in this work may provide guidance for the implementation of other complex real-time analysis in any industrial tool suite that supports a plug-in framework (for the integration of new tools) and component-based development of DRE systems.

The new release of RCM and Rubus-ICE (Version 4.0) incorporates the contributions and results presented in this thesis.

Chapter 3

Conclusions

3.1 Summary and Conclusions

In this thesis we introduced new techniques to provide a model- and component-based support for communications-oriented development of Distributed Real-time Embedded (DRE) control systems.

In order to provide a solution to the first research question, we proposed a new approach for modeling legacy network communication in component-based DRE systems. The proposed approach abstracts the implementation and configuration of communications in DRE systems. It enables the communication capabilities of a node very explicit, but efficiently hides the implementation or protocol details. Moreover, the new approach allows model- and component-based development of new nodes that are deployed in legacy systems that use predefined communication rules. The proposed approach also enables adaptation of a node when communication rules change without affecting its internal component design. As a solution to our second research question, we presented a method to extract end-to-end timing models from component-based DRE systems that are developed using above modeling approach. The purpose of extracting the end-to-end timing models is to support the Holistic Response Time Analysis (HRTA) of DRE systems.

We believe, these techniques may be suitable for several other component models for DRE systems that use a pipe-and-filter style for component interconnection. Moreover, these techniques can be used for any type of "inter-model signaling", where a signal leaves one model (e.g., a node, or a core, or a process) and appears again in some other model.

While we were looking for answers to our research questions, we identified a need for the extension of existing response-time analysis of CAN to support the analysis of common message transmission patterns that are implemented by some high-level protocols used in the industry. Accordingly, we extended the existing analysis which is generally applicable to any high-level protocol or commercial extension of CAN that uses periodic, sporadic, and both periodic and sporadic transmission of messages.

We provided a proof-of-concept implementation of our modeling and analysis approaches by extending the existing industrial component model, i.e., the Rubus Component Model (RCM); implementing the extended HRTA in an industrial tool suite, i.e., Rubus-ICE; and conducting an automotive-application case study. The analysis engines that we provide are able to predict important execution characteristics of the system such as holistic response times without a need for tedious and expansive testing.

We believe, the industrial tools that implement our modeling techniques and extended analysis for the development of DRE control systems may prove helpful for the software development organizations in the automotive domain to decrease the costs for software development, configuration and testing.

3.2 Future Work

An interesting future research direction is to investigate and develop patterns that allow transformation between several domain-specific modeling languages in the vehicular domain. The idea is to bridge the semantic gap between functional models (expressed in standard languages as EAST-ADL [21] and/or proprietary languages such as Simulink [22] or Statemate [23]) and execution models (expressed in standard languages like TADL [24] and Autosar [6] and/or proprietary languages like RCM). It would also be interesting and useful to facilitate the exchange of analysis models and tools between RCM and several other component models and tools used for the development of automotive embedded systems.

Another future work could be extending the existing analysis of CAN by combining the analysis of mixed messages in CAN (presented in this thesis) and analysis of CAN with FIFO queues [25]. The extended analysis will be able to compute the worst-case response times of mixed messages in the CAN network where some nodes use FIFO queues while others use priority queues. The preliminary work in this direction is presented in [26]. Another future work in this direction is the extension of CAN analysis for mixed messages which have multiple sources for periodic and sporadic triggering.

In the future, the HRTA plug-in can be expanded by implementing and integrating the analysis of other network communication protocols (e.g., Flexray, switched ethernet, etc.) within the holistic analysis algorithms discussed in this thesis. Another future work could be providing a support for asynchronous data-flow using the two different semantics of data-age and reaction (described in [27]) in Rubus-ICE.

Bibliography

[1] Michael Barr and Anthony Massa. *Programming Embedded Systems*. O'Reilly Media, Inc., 2006.

[2] Michael Barr. Embedded Systems Glossary. http://www.netrino.com/Embedded-Systems/Glossary.

[3] Robert N. Charette. This Car Runs on Code. *Spectrum, IEEE*, 46(2), 2009. http://spectrum.ieee.org/green-tech/advanced-cars/this-car-runs-on-code.

[4] Thomas A. Henzinger and Joseph Sifakis. The Embedded Systems Design Challenge. In *Proceedings of the 14th International Symposium on Formal Methods (FM), Lecture Notes in Computer Science*, pages 1–15. Springer, 2006.

[5] Ivica Crnkovic and Magnus Larsson. *Building Reliable Component-Based Software Systems*. Artech House, Inc., Norwood, MA, USA, 2002.

[6] AUTOSAR Techincal Overview, Version 2.2.2. AUTOSAR – AUTomotive Open System ARchitecture, Release 3.1, The AUTOSAR Consortium, Aug., 2008. http://autosar.org.

[7] M. Broy, I.H. Kruger, A. Pretschner, and C. Salzmann. Engineering automotive software. *Proceedings of the IEEE*, 95(2):356 –373, feb. 2007.

[8] Ken Tindell and John Clark. Holistic schedulability analysis for distributed hard real-time systems. *Microprocess. Microprogram.*, 40:117–134, April 1994.

[9] Robert Bosch GmbH. CAN Specification Version 2.0. Postfach 30 02 40, D-70442 Stuttgart, 1991.

[10] ISO 11898-1. Road Vehicles interchange of digital information controller area network (CAN) for high-speed communication, ISO Standard-11898, Nov. 1993.

[11] Automotive networks. CAN in Automation (CiA). http://www.can-cia.org/index.php?id=416.

[12] CAL, CAN Application Layer for Industrial Applications, CiA Draft Standard DS-207, Version 1.1. *CAN-in-Automation*, Feb. 1996.

[13] CANopen high-level protocol for CAN-bus, Version 3.0. *NIKHEF, Amsterdam*, March 2000. http://www.nikhef.nl/pub/departments/ct/po/doc/CANopen.pdf.

[14] Jimmy Westerlund. Hägglunds Controller Area Network (HCAN), Network Implementation Specification. *BAE Systems Hägglunds, Sweden (internal document)*, April 2009.

[15] MilCAN (CAN for Military Land Systems domain). http://www.milcan.org/.

[16] TIMMO Methodology , Version 2. *TIMMO (TIMing MOdel), Deliverable 7*, October 2009. The TIMMO Consortium.

[17] K. Hänninen et.al. The Rubus Component Model for Resource Constrained Real-Time Systems. In *3rd IEEE International Symposium on Industrial Embedded Systems*, June 2008.

[18] Sverine Sentilles, Aneta Vulgarakis, Tomas Bures, Jan Carlson, and Ivica Crnkovic. A Component Model for Control-Intensive Distributed Embedded Systems. In *Proceedings of the 11th International Symposium on Component Based Software Engineering (CBSE2008)*, pages 310–317. Springer Berlin, October 2008.

[19] Xu Ke, K. Sierszecki, and C. Angelov. COMDES-II: A Component-Based Framework for Generative Development of Distributed Real-Time Control Systems. In *Embedded and Real-Time Computing Systems and Applications, RTCSA 2007. 13th IEEE International Conference on*, pages 199 –208, August 2007.

[20] Arcticus Systems. http://www.arcticus-systems.com.

[21] EAST-ADL Domain Model Specification, Deliverable D4.1.1. http://www.atesst.org/home/liblocal/docs/ATESST2_D4.1.1_EAST-ADL 2-Specification_2010-06-02.pdf.

[22] Simulink - Simulation and Model-Based Design. http://www.mathworks .se/products/simulink.

[23] Rational Statemate. http://www-01.ibm.com/software/awdtools/statemate.

[24] TADL: Timing Augmented Description Language, Version 2. *TIMMO (TIMing MOdel), Deliverable 6*, October 2009. The TIMMO Consortium.

[25] Robert I. Davis, Steffen Kollmann, Victor Pollex, and Frank Slomka. Controller Area Network (CAN) Schedulability Analysis with FIFO queues. In *23rd Euromicro Conference on Real-Time Systems (ECRTS11)*, July 2011.

[26] Mubeen, Saad and Mäki-Turja, Jukka and Sjödin, Mikael. Extending response-time analysis of controller area network (CAN) with FIFO queues for mixed messages. In *Emerging Technologies Factory Automation (ETFA), IEEE 16th Conference on*, pages 1–4, sept. 2011.

[27] N. Feiertag, K. Richter, J. Nordlander, and J. Jonsson. A Compositional Framework for End-to-End Path Delay Calculation of Automotive Systems under Different Path Semantics. In *Compositional Theory and Technology for Real-Time Embedded Systems, 2008. CRTS 2008. Workshop on*, dec. 2008.