Linköping Studies in Science and Technology
Dissertations, No. 1984

Distributed Moving Base Driving Simulators

Technology, Performance, and Requirements

Anders Andersson

Linköping Studies in Science and Technology
Dissertations, No. 1984

Distributed Moving Base Driving Simulators – Technology, Performance, and Requirements

Anders Andersson

Linköping University
Department of Computer and Information Science
Division for Software and Systems
SE-581 83 Linköping, Sweden

Linköping 2019

Edition 1:1

© Anders Andersson, 2019

Thesis cover: A sketched photo of the moving base driving simulator Sim III at VTI in Linköping. Colors are from the VTI's main office facade.

ISBN 978-91-7685-090-9

ISSN 0345-7524

URL <http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-156537>

Published articles have been reprinted with permission from the respective copyright holder.

Typeset using X_YTEX

Printed by LiU-Tryck, Linköping 2019

POPULÄRVETENSKAPLIG SAMMANFATTNING

Utveckling av nya smarta system och funktioner för olika typer av fordon sker i ett högt tempo. För att garantera att dessa nya system och funktioner fungerar som tänkt så behövs flexibla och trovärdiga utvärderingsverktyg. En körsimulator är ett exempel på ett sådant verktyg för att prova/utvärdera befintliga och nya fordonskoncept och hjälpsystem. När en förare i en körsimulator kör på samma sätt som ute i trafiken får man en realistisk utvärdering av det man vill undersöka. Två fördelar med en körsimulator är att man kan upprepa samma situation flera gånger under kort tid och man kan titta på reaktioner under farliga situationer som kan resultera i allvarliga personskador ute i trafiken. En viktig komponent i en körsimulator är fordonsmodellen, dvs den modell som beskriver hur fordonet reagerar på sin omgivning och föraren. För att öka simulatorrealismen eller beräkningsprestandan kan man dela upp fordonsmodellen i delsystem som körs på olika datorer som kopplas ihop i ett nätverk. Ett delsystem kan även ersättas med hårdvara genom så kallad hardware-in-the-loop-simulering och kopplas då ihop med resten av fordonsmodellen genom specificerade interface. Metoden att dela upp en modell i mindre delsystem som körs på separata noder och kommunicerar genom ett nätverk kallas för distribuerad simulering.

I denna avhandling så undersöks om, och i så fall hur, en distribuerad simulatordesign underlättar det underhåll och nyutveckling som krävs för att en körsimulator ska kunna följa den snabba fordonsutvecklingen som sker. För detta så har tre olika distribuerade simulatorlösningar designats, byggts och analyserats med målet att konstruera distribuerade simulatorer, inklusive extern hårdvara, där simuleringen blir lika realistisk som med en traditionell körsimulator. En av dessa simulatorlösningar har använts för att skapa en parametriserad drivlinemodell för att representera ett önskat fordon. Vidare så kombineras förarens köruppgift med modellen för att övervaka modellavvikelser. I ett nästa steg så har delsystem från en simulatorlösning och drivlinemodellen överförs till en Modelicamiljö. Målet är då att genom ett automatiskt ramverk i ett verktyg garantera modellrealism, även vid distribuerad simulering.

Resultaten visar att de utvecklade distribuerade simulatorer fungerar bra överlag med tillfredsställande prestanda. Det gäller att hantera fordonsmodellen och hur den kopplas till ett distribuerat system. Provad simulatordesign gav att med ett snabbt nätverk så kan man bortse från fördröjningarnas påverkan på modellen men om man succesivt ökar fördröjningarna kommer en förare i den distribuerade simulatorn att ändra sitt beteende. Ett annat problem är hur modifieringar i systemet påverkar fordonsmodellen och det önskade beteendet. Detta leder till behov av metodik för kravhantering på modellen. För att detektera modellavvikelser i simulatormiljön så har en övervakning implementerats för att hjälpa försöksledare notera när en modell beter sig konstigt. Då tillgängligheten av distribuerad laborationsutrustning kan vara begränsad så undersöks också möjligheten att använda Modelica (som är ett ekvationsbaserat och objekt-orienterat programspråk) för att simulera delsystem. Implementering av modellen i Modelica har också utvidgats med kravhantering och i detta arbetet föreslås ett ramverk för att automatiskt utvärdera modellen i ett verktyg.

ABSTRACT

Development of new functionality and smart systems for different types of vehicles is accelerating with the advent of new emerging technologies such as connected and autonomous vehicles. To ensure that these new systems and functions work as intended, flexible and credible evaluation tools are necessary. One example of this type of tool is a driving simulator, which can be used for testing new and existing vehicle concepts and driver support systems. When a driver in a driving simulator operates it in the same way as they would in actual traffic, you get a realistic evaluation of what you want to investigate. Two advantages of a driving simulator are (1.) that you can repeat the same situation several times over a short period of time, and (2.) you can study driver reactions during dangerous situations that could result in serious injuries if they occurred in the real world. An important component of a driving simulator is the vehicle model, i.e., the model that describes how the vehicle reacts to its surroundings and driver inputs. To increase the simulator realism or the computational performance, it is possible to divide the vehicle model into subsystems that run on different computers that are connected in a network. A subsystem can also be replaced with hardware using so-called hardware-in-the-loop simulation, and can then be connected to the rest of the vehicle model using a specified interface. The technique of dividing a model into smaller subsystems running on separate nodes that communicate through a network is called distributed simulation.

This thesis investigates if and how a distributed simulator design might facilitate the maintenance and new development required for a driving simulator to be able to keep up with the increasing pace of vehicle development. For this purpose, three different distributed simulator solutions have been designed, built, and analyzed with the aim of constructing distributed simulators, including external hardware, where the simulation achieves the same degree of realism as with a traditional driving simulator. One of these simulator solutions has been used to create a parameterized powertrain model that can be configured to represent any of a number of different vehicles. Furthermore, the driver's driving task is combined with the powertrain model to monitor deviations. After the powertrain model was created, subsystems from a simulator solution and the powertrain model have been transferred to a Modelica environment. The goal is to create a framework for requirement testing that guarantees sufficient realism, also for a distributed driving simulation.

The results show that the distributed simulators we have developed work well overall with satisfactory performance. It is important to manage the vehicle model and how it is connected to a distributed system. In the distributed driveline simulator setup, the network delays were so small that they could be ignored, i.e., they did not affect the driving experience. However, if one gradually increases the delays, a driver in the distributed simulator will change his/her behavior. The impact of communication latency on a distributed simulator also depends on the simulator application, where different usages of the simulator, i.e., different simulator studies, will have different demands. We believe that many simulator studies could be performed using a distributed setup. One issue is how modifications to the system affect the vehicle model and the desired behavior. This leads to the need for methodology for managing model requirements. In order to detect model deviations in the simulator environment, a monitoring aid has been implemented to help notify test managers when a model behaves strangely or is driven outside of its validated region. Since the availability of distributed laboratory equipment can be limited, the possibility of using Modelica (which is an equation-based and object-oriented programming language) for simulating subsystems is also examined. Implementation of the model in Modelica has also been extended with requirements management, and in this work a framework is proposed for automatically evaluating the model in a tool.

Acknowledgments

First, I want to thank my main supervisor Peter Fritzson. Without him this joint work between the Swedish National Road and Transport Research Institute (VTI) and Linköping University would never have been possible. It has been a long road and I am grateful for all insightful thoughts and guidance. I also want to thank my supervisor Jonas Jansson at VTI for giving me this opportunity and for his support and valuable input. Further, I want to thank my co-supervisor Lena Buffoni who has been very helpful throughout my whole progress and I am thankful for the numerous valuable discussions and improvement suggestions throughout this work. I would also like to thank Bernhard Thiele for his valuable comments when he was co-supervisor.

Second, my work is a fusion of results from different projects. As such, there have been several financial contributors whom I am thankful to. For specific acknowledgements of the financial contributors, see each paper. I would also like to thank the Transport Research Environment with Novel Perspectives (TRENoP) framework.

My work with distributed simulators have involved cooperation with heavy equipment facilities. I want to thank Vehicular Systems at Linköping University for the collaboration around the chassis dynamometer. I also want to thank Volvo Cars and the VICTA Lab for participating with their HiL equipment. Furthermore, I want to thank all involved co-authors, and also Tobias Lindell, Erik Olsson, and Fredrik Rolff for their help in various projects.

For proofreading the kappa I want to thank, Sogol Kharrazi, my father, Mariyam Brittany Shahmehri, and Anne Moe.

I am also very grateful to all former and current colleagues at FTS/SIM and members at PELAB for a fun working environment. Many enriching topics, important and/or irrelevant, have been dealt with during fika.

Finally, I would like to thank family and friends for encouragement and support during tough times. Especially I want to thank Sogol one more time for cheering me on and keeping my spirit up when approaching the finishing line and finally completing the race. I also want to thank my son Oscar for letting me know that I need to play!

Anders Andersson
Linköping, Sweden, April 2019

Contents

Abstract	iii
Acknowledgments	v
Contents	vi
1 Popular introduction – I want a car tailored to me...	1
1.1 Research questions at large	2
1.2 My research in three parts	3
1.3 Requirements would have been useful - an example	4
1.4 Thesis layout	6
2 A brief background	7
2.1 A high fidelity driving simulator	7
2.2 A vehicle model	10
2.3 Distributed Simulation	13
3 My pieces in the big puzzle	15
3.1 Papers timeline	15
3.2 Paper A	17
3.3 Paper B	18
3.4 Paper C	19
3.5 Paper D	20
3.6 Paper E	21
3.7 Paper F	22
3.8 Paper G	23
3.9 HLA simulator demonstrations	24
4 Considerations and final remarks	27
4.1 Reflections on the work done	27
4.2 Future work	29
4.3 The work in a wider context	30
Bibliography	31

I	Distributed driving simulators	35
A	Vehicle Powertrain Test Bench Co-Simulation with a Moving Base Simulator Using a Pedal Robot	37
A.1	Introduction	39
A.2	Experimental setup	39
A.3	Results	51
A.4	Conclusions	57
A.5	References	58
B	A Driving Simulation Platform using Distributed Vehicle Simulators and HLA	61
B.1	Introduction	63
B.2	Software architecture	63
B.3	Hardware platforms	70
B.4	Results	72
B.5	Conclusions	74
B.6	Acknowledgement	75
B.7	References	75
C	Testing Cooperative Intelligent Transport Systems in Driving Simulators	77
C.1	Introduction	79
C.2	Simulators in the system	80
C.3	Systems Integration	82
C.4	Simulation Scenarios	85
C.5	Results	87
C.6	Discussion	89
C.7	Conclusion	91
II	A powertrain model and its driving task	95
D	Parameterization Procedure of a Powertrain Model for a Driving Simulator	97
D.1	Introduction	99
D.2	Powertrain model	99
D.3	Vehicle and measurement equipment	101
D.4	Parameterization methodology	103
D.5	Validation results	107
D.6	Conclusions	112
D.7	References	113
E	Vehicle model quality framework for moving base driving simulators, a powertrain model example	115

E.1	Introduction	117
E.2	Vehicle dynamics model quality framework	118
E.3	Quality assessment of a powertrain model	122
E.4	Results	130
E.5	Conclusions	133
E.6	Acknowledgements	133
E.7	References	134
III Model components and requirements in Modelica		137
F	Models for Distributed Real-Time Simulation in a Vehicle Co-Simulator Setup	139
F.1	Introduction	141
F.2	Hardware Facilities	142
F.3	Results	153
F.4	Conclusion	154
F.5	Future Work	155
F.6	Acknowledgements	155
F.7	References	156
F.8	Appendix	157
G	Powertrain Model Assessment for Different Driving Tasks through Requirement Verification	161
G.1	Introduction	163
G.2	Use case	163
G.3	Requirement model	164
G.4	Verification model	168
G.5	Conclusion and future work	173
G.6	Acknowledgment	174
G.7	References	174
Appendix		179
List of Figures		179
List of Tables		184

Popular introduction – I want a car tailored to me and my (unconscious) needs

“He never created a finished product. Finished products are for decadent minds. His was an evolving mechanism and the Second Foundation was the instrument of that evolution.”

— Isaac Asimov, *Second Foundation*

Occasionally, I rent a car for travelling during work trips. A few years ago, a new rental car introduced me to automatic cruise control. Recently, another rental car introduced me to lane keep assist. Company policy also specifies that I should rent an environmentally friendly car. During my most recent call to the rental service (November 2018), they didn’t have any cars available, since the cars that used to be eco-friendly are no longer classified as such. Current vehicle development is a rapidly changing field. Various built-in driver support systems and production systems that enable communication between vehicles, mobile devices, and infrastructure (also called Vehicle-to-everything, V2X) are entering the market. Another trend that is predicted to add many new functions/systems is the advent of self-driving vehicles. Are these new systems safe? Are they environmentally friendly? Are they understandable to a human driver? In order to investigate these questions, a moving base driving simulator can be a powerful research tool, but to be useful it must be kept up-to-date.

In this thesis, I elaborate on our attempts to improve the flexibility of moving base driving simulators while maintaining or improving realism through the use of a distributed simulator approach. In the process of presenting these topics, I have realized that people have different understandings of what a moving base driving simulator is. Since a picture is worth more than a thousand words, Figure 1.1 shows two of the moving base driving simulators used at the Swedish National Road and Transport Research Institute (VTI). These moving base driving simulators provide the foundation for this work.

Figure 1.1: The moving base driving simulators Sim3 and Sim4 at VTI.

1.1 Research questions at large

The procedure for using a driving simulator is, in short, as follows: One person, the driver, enters the vehicle located inside the black dome of the simulator and then drives the vehicle. For each driver, data is collected and later analyzed. The quality of the data is strongly dependent on capturing realistic driving behavior, i.e., test subjects must operate the vehicle in the simulator in the same way they would operate a vehicle on real roads. Ideally, a driver in a moving base driving simulator should have the same driving behavior that he or she would have when performing the same driving task in a real vehicle. With this in mind, I have focused on distributed moving base driving simulators from a technological perspective, investigating: (1) technologies for distribution; (2) the modeled vehicle and how it is driven; (3) model fidelity and requirements.

An important aspect of this work is to investigate whether or not distributed simulators can improve the realism and/or the flexibility in the simulators. In short, does a distributed simulator maintain high fidelity? In practice, can we connect one of the simulators at VTI to one or more simulator sites and derive added value from the combined simulators? Furthermore, a well-designed distributed system with clear interfaces should make it easier to change components within the simulator. Can we easily alter or change parts of the distributed simulator? And if it is easy to alter or change parts in a distributed simulator, can we design a framework for testing the simulator fidelity so that we know if sufficient fidelity has been achieved after a modification? Can we automatically monitor the simulator fidelity? Can we also extend the vehicle model fidelity test to include the driving task? Furthermore, can we use a tool that supports object-oriented modeling for our vehicle models? Can this tool also be used to evaluate the quality of the developed models? Once again keeping in mind the goal that drivers drive in the simulator as they normally do on the road.

1.2 My research in three parts

In this thesis I have grouped the included papers into three parts. Here I describe how and why they relate to my main research questions.

Distributed driving simulators

Research on distributed simulation and simulators has been performed in different fields, with military applications probably receiving the most attention. By contrast, comparatively little knowledge exists about the challenges and limitations that are present for moving base driving simulators. During my early research, I was only aware of previous work done by one person/group. At the beginning of this work it was not well known how a distributed simulator environment works, so one question that we had was whether or not it would be possible to construct a distributed moving base driving simulator that would maintain or even improve high fidelity. Major concerns were: a) if the introduced time delays would make a high-fidelity simulation infeasible, and b) if it would be possible to create fully functioning interfaces and decoupling points. Furthermore, within companies there are other heavy pieces of equipment that are used for vehicle testing purposes, for example chassis dynamometers. Therefore, it seems natural to investigate the possibility that connecting two such large pieces of equipment together would provide added value. Another important aspect of this research is intellectual property concerns with regard to equipment suppliers. Using such supplier software, there is a need for well-specified interfaces, as the software is typically run as a black box. Another alternative with distributed simulation is to run the protected software on a separate computer and then use communication techniques to perform a distributed simulation. Recent vehicle development with connected and self-driving vehicles has increased the effort that is required for evaluation and testing of concepts and functionality. To address the vast amount of required testing, distributed moving base driving simulators could provide a good testing platform. It is also interesting to try to run a simulator study using a multi-driver scenario.

One further benefit with distributed simulators with well-defined interfaces and components is the ability to seamlessly change components. This is important from the aspect of testing and prototyping for a simulator, where tests can be done in a desktop environment, then moved to a small simulator, and then later transferred to a high fidelity moving base driving simulator. This interchanging between platforms should be as seamless as possible.

A powertrain model and its driving task

It is important to ensure that high fidelity is maintained within the moving base driving simulator. One way to test our vehicle model's fidelity is to

compare it to a production vehicle. Thus, if we pick a production car, can we have the same car in the simulation so that we can see how similar the model and the vehicle are? When do we know how close our model is to a real vehicle? A vehicle model also needs to be parameterized to model a specific vehicle or a generic one. The parameterization procedure should also be fast, since it enables the possibility to model any vehicle.

When using one of the models that were developed in this work, it became clear that the driving task is tightly connected to the driving experience in the simulator, and therefore it is also connected to the requirements on the vehicle dynamics and the moving base system. The vehicle model together with the intended driving task should provide high fidelity. Also, can we incorporate the driving task into the test requirements? To ensure this, currently, the developer is responsible for manually checking the status of the model. This is time-consuming and error prone. Can we use model requirements that automatically check the vehicle model every time something is changed? Can we test both real-time properties and model accuracy?

Model components and requirements in Modelica

A moving base driving simulator consists of many systems and subsystems, which can be modified or replaced. Our focus here is the vehicle model, which is typically a mathematical description in the simulator. However, it is not uncommon to also include specific hardware using hardware-in-the-loop techniques. Designing a vehicle model is different depending on which tools are used. One way to build a component-based vehicle model is to use a modelling language/tool that supports acausal modelling. Here we investigate the usage of Modelica. A few previous works have been conducted using Modelica for driving simulators, and the question was whether or not it was feasible to also use Modelica for the vehicle model in a moving base driving simulator? Also, how does Modelica handle components that are interacting within a combined hardware and software set-up?

Can this testing be done in the same tool or environment that is being used for modelling so that a developer can test it directly? Here we tested to express requirements in Modelica to test different vehicle models.

1.3 Requirements would have been useful - an example

Requirements allow for the detection of erroneous model behavior. As such, if an error is introduced into the model that causes effects that a requirement tests for, then requirement testing would detect a problem. To illustrate how a small model change can impact a vehicle model, we present a small example from a freight train brake system. Here, first order systems of equations are used to model the brake air pressure for each train unit. Consider introducing the round-off presented in Listing 1.1. This round-off significantly reduces the

time to reach the specified end value from a lower value in this example from 350 kPa to 500 kPa in every unit in the train instead of waiting for the numerical floating point precision of the computer to reach its end value.

Listing 1.1: Added code to release brake pressure faster.

```
% If close enough to 500e3, round off to 500e3
if (p(j,i+1) > 499e3 && p(j,i+1) <= 500e3)
    p(j,i+1) = 500e3;
end
```

During normal braking, the brake pressure at 500 kPa is lowered according to what is shown in the left part of Figure 1.2. Then, the brake pressure at the last wagon starts to decrease at 2-3 seconds after the pressure in the locomotive is changed. When the round-off presented in Listing 1.1 is introduced, the response in pressure is shown in the right part of Figure 1.2. Here, the brake pressure in the last train unit starts to decrease after 15-16 seconds. This means that a round-off of 0.2 percent results in a brake response that is approximately 5 times slower. Thus, the few lines of code added in Listing 1.1 have made the train stop considerably more slowly.

A remark can be made regarding debugging this error. When dealing with an ordinary differential equation, one strategy for improving the model

Figure 1.2: Brake pressure distribution in the freight train. The left figure shows normal operation and the right figure shows the behavior when the changes in Listing 1.1 are applied.

performance is to increase the frequency of how often the equations are solved. In Figure 1.2 the simulations were done using a frequency of 60 Hz. Another common frequency is 1000 Hz. The issue is that for small steps in the solver the solution does not go below 499 kPa, and thus the pressures get stuck at 500 kPa. Therefore, after 2 minutes of simulation at a frequency of 1000 Hz, the brake pressure in the rear half of the train has not even started to change, thus the brake system has been disabled.

Our example shows that a few added lines of code result in a large error of a type that a developer might not foresee without having encountered a similar case. It also shows that if the model is run at a higher frequency to try to remedy this error, the brake system will be disabled.

In this example, the consequences of this small model error could have been detected automatically by the lack of model response to actuator signals. This is a reason for introducing model requirements that are automatically checked by a tool, so that a developer could get an automatic warning and correct the error immediately.

1.4 Thesis layout

The structure of this thesis is as follows. In the next chapter, some background information is provided, including a description of simulator fidelity, vehicle modelling, and an introduction to distributed driving simulators and requirements. Chapter 3 presents short introductions to the thesis contributions within this field. In Chapter 4, reflections on the contributions and ideas for future work are provided. The selected papers have been appended at the end.

A brief background

Captain Lennox: NBEs?

“Non-Biological Extraterrestrials. Try to keep up with the acronyms.”

— Agent Simmons, *Transformers*

In this chapter, a brief background is provided on concepts that are commonly used in this thesis for the benefit of readers who are not familiar with moving base driving simulators. First, we provide a definition of what a moving base driving simulator is, when it has high fidelity, and examples of current simulators at VTI. Second, the difference between causal and acausal modelling is discussed, including tools and languages for acausal modelling, and how they aid an object-oriented development. Third, we look into model requirements and how the driving task affects the validity of the vehicle model. Finally, we provide an introduction to distributed simulation and hardware-in-the-loop simulation.

2.1 A high fidelity driving simulator

Simulations and simulators are common today. Driving simulator tools range from purely mathematical models running on a single computer to full size moving base driving simulators that include several computers and take up the physical space of a small hangar, e.g. the National Advanced Driving Simulator [25]. Most driving simulators fall somewhere between these extremes, and this work focuses on high fidelity moving base driving simulators.

Moving base driving simulator characteristics

A moving base driving simulator has a few distinctive features.

1. A moving base that gives motion cues to the driver.
2. A realistic cabin that permits a driver to interact with the simulated vehicle.
3. A visual and auditive (sometimes also haptic) representation of the surrounding environment.

A moving base driving simulator with only one or two degrees of freedom is referred to as a small moving base simulator. Thus, what is referred to as a moving base driving simulator will have three or more degrees of freedom. How these degrees of freedom are used varies depending on the simulator and its application. The strategy for providing motion feedback to the driver is called motion cueing. For an example of a motion cueing strategy see [15].

These distinctive features only describe the top layer of a moving base driving simulator. Other important technical aspects include capabilities for logging of data, a vehicle dynamics model, models for surrounding traffic behavior, and digital environments representing actual roads/cities or totally fictional environments. There can also be additional study-dependent equipment, such as systems for logging biological signals or systems for driver distraction tasks. Sometimes the system or component being tested is an actual piece of hardware that is run as hardware-in-the-loop.

High fidelity

The term “high fidelity” was originally used to describe equipment that reproduces sound. In this setting, high fidelity equipment reproduces the original sound accurately, e.g., with correct frequencies and low noise. For driving simulators we consider high fidelity to be tightly connected to the driver behavior and experience. Usually, the goal of a simulator is to study driver behavior during different situations, and thus it is crucial that the driver is immersed in a normal driving behavior. As such, a high fidelity driving simulator fulfills what we define here as the *simulator driver behavioral validity* (generally relative validity).

Simulator driver behavioral validity: *A driver in a moving base driving simulator should have the same driving behavior as he or she would have when performing the same driving task in a real vehicle.*

Absolute validity, where there is no statistical difference between driving in a simulator and driving on the road, is a hard goal to achieve [35]. Thus, usually the aim is to achieve behavior validity for specific parts of a simulator study. One such example is a study [1] performed at the Swedish National Road and Transport Research Institute (VTI), where the goal of absolute validity was achieved for parts of the study. In this study a lot of effort went

into preparing the simulator, because it involved having drivers alternate between driving on actual, physical roads near VTI, and then driving on these same roads in the driving simulator. Thus, the simulator environment had to match the outdoor environment with regard to, e.g., landmarks. Also, the vehicle model within the moving base driving simulator was parameterized to match the specific vehicle that was being driven on the road. For certain applications where a specific driving maneuver is known or expected to occur, the simulator can be optimized for it beforehand in order to get closer to absolute validity. For example, the simulator can be prepositioned for curves, see [19], although most cases of normal driving do not allow for such optimizations. Since it is often not possible to put this amount of effort into constructing the environment or calibrating the simulator, the goal for driver perception would be to have relative validity. This means that driver behavior in a driving simulator is relatively correct compared to drivers on the road. For example, consider a driving simulator study comparing two different designs of an active safety system that alerts the driver. Assume that one of the systems results in a lower reaction time when compared to the other system for a specific situation. Then, if the simulation is relatively valid, the same system will have a lower reaction time compared to the other system in real driving as well. Improving the simulation immersion and perception increases the precision in driving perception and thus aids the validity of the results, but it has to be done in a cost-effective way.

Driving simulators at VTI

At VTI, moving base driving simulators have been used and developed for more than thirty years, starting in 1978 with the presentation of the first construction plans for Sim I, which was later completed in 1984. Sim I had a moving base, a visual system, a cabin and a model of a vehicle [26] but was dismantled several years ago. As simulator development continued, it became quite clear that the simulators and the studies they perform are quite connected and that there will always be limits to what is possible [27]. A common study performed using the VTI moving base driving simulators focuses on driver behavior. For a few examples see an early collection of behavior research [26], a study of mobile phone usage [21], a study about detection of sleepiness [16] and a study evaluating steering feeling in heavy vehicles [34]. However, studies at VTI are not solely limited to behavioral studies, and several other aspects have been investigated e.g. dynamic performance of trucks in a collision avoidance maneuver on low friction surfaces [22] and visualizations of construction plans for a tunnel [32].

Currently there are three operational moving base driving simulators at VTI which are:

Sim II: *Sim II was constructed in 1991 to be used as a truck simulator [28]. The simulator provides 3 degrees of freedom for larger motion and a vibration table for road roughness, and has been upgraded in several steps improving, e.g., the visual and audio system.*

Sim III: *Further developing the design used in Sim II, one more degree of freedom was added when constructing Sim III, which was finished in 2003 [30]. The simulator is mostly used as a passenger car simulator with a smooth linear motion with world class performance.*

Sim IV: *Sim IV was built with a new design combining a hexapod with an xy-table, providing large linear motion in the plane [20]. The simulator was inaugurated in 2011 and is used with a car or a truck cabin.*

In addition to the moving base driving simulators, smaller fixed base driving simulators are also used at VTI.

Although the work done in this thesis focuses on road-bound driving simulators, it might be of interest to mention that VTI also develops railway train simulators. To reduce the amount of different software and hardware systems, an architecture structure in which the train simulators share as much as possible with the driving simulators is desired. An example of hardware interoperability is that the dimensions of one train simulator cabin are such that it is possible to mount the train cabin onto the moving base in Sim III.

2.2 A vehicle model

A model of a physical system should capture essential characteristics for an intended usage. Furthermore, in a broad sense a simulation demonstrates how such a model behaves over time. As such, the name “driving simulator” specifies the usage of a physically modelled vehicle and its behavior over time. Here we take a further look into such vehicle models, how different parts of the models can be interchanged, including usage of hardware, and how requirements can be imposed on the models.

Acausal modelling

Physical modelling of the vehicle will typically use a representation of differential algebraic equations (DAE) combined with discrete events, e.g., when a driver changes gears, and state machines. Modelling such systems can be done in a number of different ways using many different tools. One straightforward method would be to use a procedural programming language, such as Fortran or C/C++, and let an engineer design how the equations will be solved by implementing a solver for the DAE system. Another approach would be to use a tool that provides a graphical user interface that allows the engineer to specify the flow of information in an input/output manner. One example of

such a tool is Simulink, which is made by MathWorks. Here the task for the engineer is simpler, since Simulink provides the numeric solver. However, in this type of approach the direction of information flow is embedded into the model, which can be a drawback. When the engineer has to manage the order in which equations are solved we have causal modelling. Another approach is to reduce the manual work by letting the engineer write the equations directly and let the compiler sort the equations automatically, then use a numerical solver to solve the resulting system, e.g., DASSL which is a numerical solver for implicit systems of differential and algebraic equations [33]. This kind of modelling is called acausal modelling.

Modelica

One example of a modelling language that handles acausal modelling is the equation-based language Modelica, see e.g., [17] or [23], which is based on an open standard and has extensive tool support. Modelica lets the modeler write acausal models, including hybrid modelling (continuous and discrete time), and also includes a graphical user interface with a rich standard library of components. Modelica supports object-orientation, wherein a large model may be built from components, which provides a high level of reusability. For comparison, consider the example of Simulink, where the internal description of the model is not freely available and thus the only tool that can handle the model is Simulink itself. As Modelica is a wide-spread language specification there are several different tools available that can transform a model into computable code, thus the user can freely choose a tool depending on the application. To give an example of the modelling methods we have described, consider Table 2.1, in which two implementations are presented, one in C++ and one in Modelica. For the system that is modelled in C++ the model will only solve for $\mathbf{h}$, while for the Modelica model the system could be changed by the compiler to be solved for, e.g., $\mathbf{Q_out}$ if $\mathbf{h}$ was given, depending on how the engineer uses the model. Relieving the developer of practical details such as implementing a solver for the DAE makes acausal modelling attractive.

In this work, the approach has been to increase usage of Modelica for the vehicle model. One free and open source tool for Modelica is OpenModelica developed at Linköping University together with the Open Source Modelica Consortium [31]. OpenModelica supports the whole standard and the Modelica Standard Library, and also has several extra features such as support for compiling Functional Mock-up Units (FMUs) according to the Functional Mock-up Interface (FMI) standard [10], a graphical user interface, and automatic parallelization with ParModelica.

Table 2.1: A one tank system implemented in C++ and Modelica omitting routines for initialization and set and get input and/or output.

Modelica	<pre> model tank input Real Q_in "flow in"; output Real Q_out "flow out"; parameter Real A "tank area"; Real h "water level"; equation Q_in - Q_out = A * der(h); end tank; </pre>
C++	<pre> class tank { public: updateLevelExplicitEuler(double dt) { h = h + dt*(Q_in - Q_out)/A; } private: double Q_in; //flow in double Q_out; //flow out const double A; //tank area double h; //water level } </pre>

Automatic model verification

One important aspect regarding modelling of rapidly evolving or changing systems, such as vehicles, is the model evolution over time. In this aspect a model of a vehicle needs to evolve to keep a high level of fidelity over time. Also, for future use it is good to consider modelling tools that are expected to respond well to changes within a dynamics model such as changing or modifying a submodel. In software development, unit testing is used to test software changes so that the software still maintains certain functionality. Here we try to achieve similar testing functionality by evaluating formally stated requirements when changes are made to the vehicle model. Thus, alongside a vehicle model we also execute a requirement model. An evaluated requirement can have one of the following values: `VIOLATED`, `NOT_VIOLATED`, or `NOT_APPLICABLE`. The general idea is that if every requirement after a simulation still has a state of `NOT_VIOLATED`, the model is still valid. This type of testing is preferably done before a study, preferably directly when running a desktop simulation, but can also be done in real-time during a simulator study. For example, it is desired that a tool such as OpenModelica should provide the user with direct information from an initial simulation if any changes that have been made are accurate or if model fidelity has deteriorated too much. In cases where the same modelling tool is used, both for the vehicle model and for the testing, the requirements check can be run directly by the developer.

Also, if development has been done over several years with a rich history, there is a need to handle legacy code. As an example of how legacy code can evolve, consider the early vehicle model at VTI that was programmed in

Fortran [29]. Over the more than thirty years of development, other programming languages have been introduced into the simulator software with the majority being C and further C++, Matlab, Simulink, Modelica, and XML with available libraries such as Boost, and Qt. The result is a heterogeneous simulator system with legacy hardware and software which can be challenging to test.

2.3 Distributed Simulation

Distributed simulation is a technique in which different parts of a simulation at physically different locations are connected together over a network. Connecting systems over different networks increases the complexity of the simulation, since the added communication has to be managed. One major benefit of a distributed simulation is that systems at physically different locations, which might not be possible to move, can be connected. Thus, a distributed simulation might be the only way of performing certain simulation studies that require specific accurate system setups. For example, consider the setup presented in Paper A, in which a moving base simulator and chassis dynamometer laboratory are connected to each other. In such a case it is not feasible to move either the moving base simulator or the chassis dynamometer. A negative aspect of distributed simulation is that delays are introduced, which may be undesirable. The combination of distributed systems with real-time requirements presents a particular challenge, and techniques have been developed to deal with such setups in [18] and [13].

In this work, three different approaches (papers A, B, and C) have been investigated to create a distributed simulation combined with HiL, including moving base driving simulators. Different simulators have different purposes, and different strategies have been tested to see how well different simulators can be combined in the existing literature. However, limited research exists on distributed simulation combined with moving base driving simulators. At VTI only one study has made a prototype implementation [36]. One of the few examples of a distributed simulation, in which connections were made over the internet combining an engine test lab and a moving base driving simulator, can be found in [11] and [14]. Distributed simulation has been used in other fields, where for example, the IEEE standard High Level Architecture (HLA) has been used extensively for distributed simulations in military applications [24]. An example where an HLA compliant design was used for a vehicle simulator can be found in [12].

There can also be different subcomponents of a physical vehicle model that are simulated in a distributed platform. In such a case, there is a tighter limit on the time delay that can be tolerated. With regard to the update frequency in a driving simulator, the requirements at VTI are rather strict.

It is common to use a dynamics model update frequency of 1000 Hz, which is necessary in order to provide a good steering feel.

Hardware-in-the-loop

Hardware-In-the-Loop (HiL) is a method in which hardware is interfaced into a software controlled environment/simulation using sensors and actuators. The purpose of introducing HiL can be either to test the hardware system or to provide extra fidelity to a software simulation. The closer the test hardware is to the final hardware production system, the higher the probability is that the HiL system will be correctly perceived and evaluated. HiL is a very general term and the inclusion of hardware does not specify the degree to which hardware has been used. Other terms are sometimes used to further specify what is meant by HiL. In this thesis, such further specifying terms are avoided, and HiL is used instead with a thorough description of the setup. One important difference from other types of simulations is that a HiL simulation has real-time demands due to hardware response times. Thus, it can be difficult to do a simulation faster than real-time, e.g., batch simulations and restarting a simulation can be time-consuming since hardware requires time to get into a correct initial state.

For a moving base driving simulator, the ability to add hardware into the simulation is rather essential. For instance, consider the usage of a cabin which not only has the steering wheel and pedals but also contains a connection to the internal car CAN bus for in-vehicle buttons. Moreover, it is advantageous to use a moving base simulator with HiL, since it provides faster design cycles and better fidelity for the driver without the need for a prototype vehicle. HiL simulation requires an interface to ensure functionality between the hardware and the software model. This is essential if the vehicle model has parts/submodels that are black box models that need to interact with the hardware or other vehicle model parts.

My pieces in the big puzzle

We begin here by giving a timeline progress of the work that we have performed and how the included papers are connected to each other, see Figure 3.1. After that, we continue with short descriptions of the most important aspects of the included papers. Finally, we present a section with continued work on the framework in Paper B.

3.1 Papers timeline

When we started this work, there weren't many publications or applications for a distributed platform for moving base driving simulators. We begun by investigating whether it was possible to design and create such a setup connected to a large-scale HiL laboratory. Results from these efforts are presented in paper A, where a co-simulation of a driving simulator and a complete vehicle powertrain is presented. Using this setup, it was realized that it is not always possible to have the chassis dynamometers laboratory available at the same time as a moving base driving simulator at VTI. Thus, work continued with the creation of mathematical models with the same interface as was used in the co-simulation setup. The idea is to be able to use either software models or hardware systems interchangeably, and this work is presented in paper F. Also, it became clear that by using the limited set of available sensors in the chassis dynamometer setup and an OBD sensor it was possible to parameterize the powertrain models that were being created. This enabled a better sense of how similar a vehicle model was compared to the vehicle it modelled in the chassis dynamometers. Paper D presents the initial parameterization procedure. The procedure and powertrain model used were later developed further by another master's thesis student and these models were used in the work in Paper E and translated to Modelica in Paper G.

Figure 3.1: A timeline that shows the process from when work within a project started until the resulting paper.

A distributed setup needs to maintain sufficient fidelity. To begin to explore what is meant by “enough fidelity”, requirements were formulated in Modelica with the intention that a model could be tested according to the requirements to guarantee sufficient fidelity. The first work in this direction is presented in paper G, where the focus is on the vehicle model. One shortcoming of this approach was the lack of the connection to the driving task that the driving simulator was designed to facilitate. This means that a vehicle model may not have to be designed to manage every possible situation rather, it needs to realistically represent the current driving task. A framework for evaluating a vehicle model’s fidelity and assessing its fitness for a specific driving task is presented in paper E.

In parallel to the work described in these included papers, two other platforms for distributed simulation have been developed. First, a distributed simulation platform based on HLA was developed, see Paper B. Extensions to this platform are also presented in Section 3.9. Second, a platform for cooperative intelligent transportation systems (C-ITS) was developed and tested. This platform is presented in Paper C.

To summarize, the presentations of our work are organized so that papers A, B, and C focus on prototype implementations for distributed moving base driving simulators. Papers D and E focus on vehicle modelling and the problem of creating requirements to permit automatic evaluation of a model’s fidelity. Finally, in papers F and G we look at implementing these concepts in Modelica using OpenModelica as a tool.

The author is the main contributor to all of the included papers except for Paper C. In Paper C, the author mainly made significant contributions in the connection between Sim IV and the SCALEXIO. In the extended work presented in Section 3.9 the primary work is done by the author.

3.2 Paper A

Figure 3.2: Chassis dynamometers set-up at Linköping University on the left and the driving simulator Sim III at the Swedish National Road and Transport Research Institute on the right.

Paper A presents a distributed co-simulation, connecting the moving base driving simulator Sim III at VTI to a vehicle connected to a chassis dynamometer at the Vehicular Systems laboratory at Linköping University, a distance of approximately 500 m. The paper presents how a pedal robot with actuators and sensors provides an interface between the driver in the simulator and the car. The fast communication of the platform enables the driver in the moving base driving simulator to drive the simulator vehicle by using a powertrain from a production vehicle positioned at the laboratory at Vehicular Systems.

At VTI, the purpose of this effort was to investigate the possibility of improving the driver perception of the powertrain in the moving base driving simulator using a distributed simulator setup. This first prototype achieved good and promising results, and with further potential improvements it was concluded that adequate fidelity could be achieved for a realistic simulator case study. These promising results encouraged the further development of distributed simulators. It was also the first time a car in a chassis dynamometer was driven by a driver in a moving base driving simulator.

3.3 Paper B

Figure 3.3: Schematic view of the distributed driving simulator architecture using HLA.

In Paper A, the communication layer between the distributed systems is created from scratch. Communication frameworks for distributed simulations already exist, so we extended a framework of this type for our distributed moving base driving simulator. In Paper B, the usage of the IEEE standard HLA Evolved is investigated. A distributed simulation architecture was designed and implemented, which divides a driving simulator environment into four major entities with well-defined interfaces, using HLA Evolved as the method of communication.

Results shows that original functionality was maintained while increasing flexibility, going from a smaller to a larger simulator, and adding scalability and the ability to add simulators to the same environment. The architecture provides clear interfaces between simulation entities. It was shown that approximately one millisecond of overhead latency was added when using HLA, which was considered to be acceptable for the graphical system but needs more investigation for the interaction between the driver and the vehicle dynamics.

3.4 Paper C

Figure 3.4: Simulator setup using the moving base driving simulator Sim IV, a SCALEXIO hardware-in-the-loop simulator, the traffic simulation tool SUMO and a communication network between traffic actors called plexe-veins. The figure shows the setup that was used, where the traffic, network and driving simulator was located at VTI Sim IV. The HiL simulator at VICTA lab was connected to VTI using a VPN connection.

In a future that includes highly or fully automated vehicles, one type of functionality that is expected to be important is connected vehicles and road infrastructures with the ability to communicate with each other. Connected and automated transportation systems share certain features with distributed simulators so distributed simulators should be a promising tool for creating realistic scenarios for testing such complex systems. In this work, a distributed simulation platform was created based on existing available components. These components are: the moving base driving simulator, SimIV, at VTI; the real-time HiL simulator based on a SCALEXIO Rack System, at VICTA Lab; and the traffic and communication simulator Platooning Extension for Veins. Here, co-simulation of these three different simulators has the potential to enable each simulator to demonstrate its strengths so that the integrated environment could realistically create possible test scenarios for C-ITS.

The simulator platform we developed was successfully tested using two scenarios, first a cross section scenario and later an overtaking scenario on a curved road. During the overtake scenario the simulated network performance was intentionally modified during the scenario, e.g., increasing latency. Thus, the simulator platform had a means of utilizing a moving base driving simulator with a HiL simulator together with a simulation of surrounding traffic, including communication between vehicles. The conclusion was that the simulator platform is capable of testing a connected and cooperative traffic system including its vehicles.

3.5 Paper D

Figure 3.5: Validation run of a parameterized powertrain model for a front wheel drive Golf V. Parameterization only uses data collected from the chassis dynamometers and the car on-board diagnostics.

One challenge when constructing a physical system with a model thereof is that the model needs to be parameterized, preferably using the same physical system. For example, there can be large differences between a generic car model and a specific car model. In Paper D, a rapid and non-invasive way to model and parameterize powertrains of test cars is explored. The powertrain model and parameterization procedure that we developed uses the chassis dynamometers setup presented in Paper A. Furthermore, the parameterizing procedure uses the on-board diagnostics of the car and does not require any additional invasive sensors. An important objective in paper A was the ability to be able to change cars easily in the chassis dynamometers laboratory. Leveraging the ease of changing cars and the data that was collected from the limited set of sensors, a parameterization procedure was developed to quickly and easily create a vehicle dynamics powertrain model that represents the current vehicle in the setup.

The parameterizing procedure was used to model a front wheel drive Golf V with a 1.4L multi-fuel engine and a manual gearbox. The results show a good match between simulation results and test data. The powertrain model has also been tested in real-time in driving simulator Sim III with satisfactory performance.

3.6 Paper E

Figure 3.6: Collected driving data for two different routes in Linköping. Data contains sections with rural driving, lightly trafficked city driving, and motorway driving.

Reliable results from a simulator study depend strongly on the driver's perception of how realistic the simulator feels while performing the driving task, see the Simulator driver behavioral validity definition in Section 2.1. In paper E, we present an automated framework that detects possible fidelity problems in the simulated longitudinal dynamics of a powertrain model in a moving base driving simulator, both prior to running a simulation and in real-time during a simulation. The framework consists of a vehicle system model, developed and parameterized in paper D, and a quality model, which depends on the driving task, which run in parallel in real time.

Results show that the powertrain model that is developed in paper D is accurate enough for a driving scenario on rural roads/motorway, but need improvements for city driving. This was expected, considering the complexity of the vehicle dynamics model, and it was accurately captured by the proposed automated framework, which also provides real-time information to the simulator operator.

3.7 Paper F

Figure 3.7: Mixed environment with interchangeable component interfaces in Modelica and hardware.

The co-simulation setup described in Paper A requires that both the moving base driving simulator and the chassis dynamometer are available when developing or testing. If the complete co-simulation setup is not available, but parts of it are, it might still be desirable to make experimentation possible. Within this scope, Paper F investigates the potential for using Modelica for physical modeling of subsystems in such a distributed simulation. The objective is to be able to interchange software with hardware-in-the-loop, e.g., using either the chassis dynamometer or a model thereof.

In this work, a Modelica car powertrain model of appropriate complexity was created and parameterized using data collected in the chassis dynamometers. The powertrain model was tested using OpenModelica and Dymola. The Modelica models that we developed were then further evaluated using different solvers, and are estimated to have real-time performance given a good run-time environment. The simulation time for one time step of the powertrain model was measured at 0.2 %, which indicates a broad margin for real-time. Thus, the potential for using Modelica for vehicle models in a distributed simulator was seen as promising work to continue.

3.8 Paper G

Figure 3.8: The left figure pictures a simple finite state machine used for a requirement. In the right figure, four requirements in Modelica are automatically verified for two different models and two different roads.

Before conducting a simulator study, it is important to assess the suitability of the vehicle system model. This model assessment should preferably be as automated as possible to decrease the need for manual testing. Paper G presents a methodology based on finite state machine requirements to verify system model behavior in a Modelica environment, where the system model is meant to be used in a moving base driving simulator. The methodology is illustrated with a use case of a Modelica powertrain system model with replaceable components and measured data from a Golf V. The Modelica powertrain model is an extended and improved model using model parts developed for Papers D and F.

Results show that automating parts of the requirement verification process helps the system user. If system issues arise, the framework helps pinpoint where a human expert should perform a detailed analysis of the simulation results in the time frame surrounding the violation. Simulating the requirement model alongside the system model adds to the total simulation time. The additional overhead here was negligible due to the small size of the requirement model, but for a larger model it might be necessary to account for this.

3.9 HLA simulator demonstrations

Figure 3.9: Simulator setup used during demonstration of the developed HLA-based simulation architecture in Gothenburg. SC, or Session Control, is controlling the simulation. ES, Environment Simulator, contains the road with its surrounding environment and vehicles. Sim IV, Desktop and VCC HiL are three different simulators.

The work in Paper B was extended further and was presented during two demonstrations with moving base driving simulators, one in Linköping and one in Gothenburg. In Figure 3.9 the setup used during the demonstration in Gothenburg is shown, where the connection from Volvo Car Corporation (VCC) to VTI was established via a mobile phone due to VPN issues. The setup used during the demonstration in Linköping was similar to the one in Gothenburg, except that the Sim IV and the VCC HiL simulators were replaced by Sim III and Sim II.

During both demonstrations, three vehicles, connected to different driving simulators, were driving within the same simulation scenario. At the demonstration in Gothenburg the vehicles were connected in the following way: one of the vehicles was driven by the driver in Sim IV with the moving base active another one was controlled from the VCC HiL simulator, and the last vehicle was driven via a desktop simulator with a gaming steering wheel setup. The scenario started with the driver in Sim IV driving the leading car and the driver in the VCC HiL driving as a second car following the leader. The driver in the VCC HiL could choose to follow the lead vehicle by either driving manually or by using an automatic cruise control (ACC) function in hardware. This demonstrated how a driver in Sim IV could interact with the driver in the VCC HiL simulator. During this part of the scenario the driver in the desktop simulator, controlling the third car, was driving around freely. After the car-following part of the scenario, the three vehicles were allowed to drive freely together. During the free driving part of the scenario the moving base

of Sim IV was turned off so that more participants at the demonstration could try the setup, since starting and stopping of the moving base takes time. In Linköping a similar setup, but without the ACC hardware, was used with the simulators Sim II, Sim III and a desktop simulator.

The demonstrations showed working implementations of distributed simulations, including hardware-in-the-loop and fully operational moving base driving simulators. From these demonstrations, the following two conclusions could be drawn. Firstly, the simulator systems should be robust, such that every participating simulator in a simulation must operate without problems. Here, every simulator that is added increases the risk that problems will occur, so adding a non-robust simulator will probably halt the complete simulation fairly often. Secondly, it is important to have good insight into the network topology that is being used, since hardware equipment at different companies can be protected by, e.g., firewalls, and can therefore be hard to reach. For further information about the demonstrations, see [2].

Considerations and final remarks

“Figuring things out for yourself is the only freedom anyone really has. Use that freedom. Make up your own mind, Rico.”

— Jean Raszczak, *Starship troopers*

My PhD process have been progressing over a number of years. During this time, I’ve had the benefit of seeing the field develop. So, in retrospect here are my reflections on my contributions.

4.1 Reflections on the work done

We begin with Paper A. In our early work we wanted to see if we could improve the fidelity of the simulator by connecting it to a vehicle powertrain. The paper shows that the setup worked remarkably well, even for an early prototype setup. One of my takeaways in this work was that the communication delay was small compared to delays in the powertrain. This means that for many applications the communication time can be ignored. As such, I envision applications where it would be possible to get some benefit from connecting heavy stationary equipment which is difficult to move to a moving base driving simulator. As the communication was several orders of magnitude faster than the measured powertrain dynamics, a setup with delays that were 100 times longer would probably function as well. In other words, it would be possible to connect equipment in different cities.

In parallel, work was conducted for Paper F. Here, one challenge was development of device drivers for connecting to hardware. I developed an external function for Ethernet and UDP handling, but they were only available for Windows, not Linux. For use with OpenModelica, it is also necessary that any external software that is developed is able to be linked into the Open-

Modelica environment. Recently, a device driver library has been developed. It would be interesting to redo some parts of this paper with the addition of these libraries, which would make it possible to further test the device libraries and investigate the relative ease or difficulty of changing components between hardware and software.

The next step in Paper D is something I consider important. One question that came up was: How accurate is current vehicle model? When making small changes to the model, is it still accurate? These are still very relevant questions for distributed simulation, I think. When we distribute, we introduce time delays and other aspects that can deteriorate the quality of the model output. As such, it is important to try to get a baseline data set, which was addressed in this paper. This data was also used to construct different versions of a powertrain model which has been very useful.

During the time around 2015, a lot of work was done on the project SimArch2, which is presented in Paper B and Section 3.9. For this project, specific tools were used for the creation of a full-scale HLA implementation for the driving simulators. The results of this work provided a robust communication link, so that when demonstrating the work over a troublesome network configuration, everything worked smoothly. The downside of the HLA implementation was that the tools used were costly for VTI. This work also demonstrated that the ability to go from a small-scale simulator to a full-scale moving base simulator is valuable, and it has been continued at VTI by further modularizing our software. A recurring question I have been asked is, “When creating a distributed simulation, do you need to base it on a bigger framework/standard or not?”. For a smaller project (one or two participants) I generally advise against since it is not worth the extra costs for tools and the added complexity. Instead, it would probably be faster to just adapt the current code to the needs of the specific project. For a large project with 3 or more contributing partners, a standard really helps.

Also, from our work in Paper A it was quite clear that if the added time delays from the distributed platform can be neglected, constructing a distributed simulator works well. Big time delays will result in worse performance, where the driver clearly notices the effects from the distributed connections. This leads to questions regarding where the limit is. How much delay can we handle? The answer to this question depends on the expected driving scenario and the parts that will be distributed. For example, in Paper A it was estimated that calm motorway driving would work well. I estimate that for certain driving tasks it is tolerable to have quite a large time delay. This led us to the requirements work. When we have a vehicle model, we set requirements on how fast it needs to deliver a numerical solution. With requirements, it is possible to test a vehicle model’s accuracy and how a time delay affects it. First we have a test without using Modelica, which is presented in Paper E. A test that uses Modelica requirements state machines can be found in Paper G.

The most recently developed distributed simulator platform at VTI is designed to investigate C-ITS, see Paper C. Here, we constructed a distributed simulator platform by combining a HiL simulator with the moving base driving simulator, a network simulator, and a traffic simulator. The method of using already-available software components and connecting them together is a good way forward I believe, and in my experience significantly reduces the implementation efforts compared to creating one's own implementations.

4.2 Future work

For future work, my first step would be to look further into requirements that use a combination of the vehicle model and the driving task. My aim would be to have a single tool that would allow a model developer to create, modify, and test vehicle models for distributed simulator applications. Initial work done in Paper E, with the combined dependency on the driving task and the vehicle model, and G, with the Modelica requirement framework, provides a basis to continue building on. I would also like to extend this basis with more support for adding network elements into the simulation, e.g., enable testing of a model with network delays. I think such a tool platform would be helpful when investigating vehicle model performance for distributed simulators.

Another topic that needs to be investigated further is the simulator driver behavioral validity. This investigation should focus on quantifying the simulator driver behavioral validity with higher precision. The goal would be to find values for limits for when a driver drives as he or she normally does, which could then be incorporated into the requirement framework. The more accurate the requirements are, the more accurate the evaluation of the vehicle model during its driving task will be. For example, when investigating C-ITS functions within a transport system, I believe that a good tool for distributed simulators and an accurate notion of fidelity will be important.

Related to the Modelica environment, I have had problems with debugging. When there is something that does not work it can be difficult to find where the error is. The larger the model complexity, the larger the effort to find the error. This is something that has improved substantially during the time of this thesis, but I think there is still a need for improvement. I think one of the stronger points of Modelica is the component view, which enables a flexible vehicle model. To be able to exploit this strong point it is important to get valuable debugging information when you try to insert a component that happens to have errors. I think this is the way forward, since I think the combination of specialized submodules/components is the way forward.

4.3 The work in a wider context

The transport system plays a very important role in our everyday life. Society as we know it would not exist without our current means of transporting both people and goods. As such, this is something that needs to function smoothly today, tomorrow, and probably for a very long time into the future. The transport system needs to be safe and environmentally friendly, among many other things. Current technical systems in vehicles have improved, but they rely on the driver being responsible for how the vehicle is operated. For automated vehicles, this will shift so that the responsibility for how the vehicle is driven is expected to be shared between the driver and the production company. To ensure that the vehicle's systems provide a benefit in a safe and secure way, the need for testing is great. This work provides a platform for further testing and evaluation of vehicle systems from (mostly) a driver's point of view. My research provides some insight into the requirements for setting up testing equipment and how it can be used to provide answers. Also, my studies on requirements on vehicle models is a starting point for more robust and automated model verification and its integration into a single tool.

Bibliography

- [1] C. Ahlström, A. Bolling, G. Sörensen, O. Eriksson, and A. Andersson. *Validating speed and road surface realism in VTI driving simulator III*. VTI rapport 745A. Swedish National Road and Transport Research Institute, 2012.
- [2] A. Andersson, J. A. Hultgren, R. Leandertz, M. Johansson, S. Betnér, O. Jakobson, and F. Rolff. *SimArch 2: Implementation and demonstration of the SimArch architecture*. Tech. rep. ViP publication: ViP - Virtual Prototyping and Assessment by Simulation, 2016-2, 2017.
- [3] A. Andersson, J. A. Hultgren, R. Leandertz, M. Johansson, S. Eriksson, and O. Jakobson. “A Driving Simulation Platform using Distributed Vehicle Simulators and HLA”. In: *Driving Simulation Conference 2015. 16-18 september 2015, Tübingen, Germany*. 2015, pp. 123–130.
- [4] A. Andersson and L. Buffoni. “Powertrain Model Assessment for Different Driving Tasks through Requirement Verification”. In: *Proceedings of The 9th EUROSIM Congress on Modelling and Simulation, EUROSIM 2016, The 57th SIMS Conference on Simulation and Modelling SIMS 2016*. 142. 2018, pp. 721–727.
- [5] A. Andersson and P. Fritzson. “Models for Distributed Real-Time Simulation in a Vehicle Co-Simulator Setup”. In: *Proceedings of the 5th International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools* (2013).
- [6] A. Andersson and S. Kharrazi. “Vehicle model quality framework for moving base driving simulators, a powertrain model example”. In: *International Journal of Vehicle Systems Modelling and Testing* 13.2 (2018), pp. 93–108.

- [7] A. Andersson, S. Kharrazi, S. Lind, and A. Myklebust. “Parameterization Procedure of a Powertrain Model for a Driving Simulator”. In: *Advances in Transportation Studies* 1 (2016).
- [8] A. Andersson, P. Nyberg, H. Sehammar, and P. Öberg. “Vehicle Powertrain Test Bench Co-Simulation with a Moving Base Simulator Using a Pedal Robot”. In: *SAE International Journal of Passenger Cars - Electronic and Electrical Systems* (2013).
- [9] M. Aramrattana, A. Andersson, H. Burden, F. Reichenberg, and N. Mellegård. “Testing Cooperative Intelligent Transport Systems in Driving Simulators”. In: *Submitted to: Transportation Research Part F: Psychology and Behaviour*. 2019.
- [10] T. Blochwitz, M. Otter, and M. Arnold. “The Functional Mockup Interface for Tool independent Exchange of Simulation Models”. In: *Proceedings of the 8th Modelica Conference*. Dresden, Germany, 2011, pp. 105–114.
- [11] M. Brudnak, M. Pozolo, M. McGough, T. Mortsfield, A. Shvartsman, and R. Romano. “Motion Base Simulation of a Hybrid-Electric HMMWV for Fuel Economy Measurement”. In: *SAE Int. J. Commer. Veh.* 1.1 (2008), pp. 41–53.
- [12] H.C. Chao, T.W. Pearce, and M.J.D. Hayes. “Use of the HLA in a Real-Time Multi-Vehicle Simulator”. In: *Proceedings of the CSME Forum*. 2004, pp. 1–4.
- [13] T. Ersal, M. Brudnak, A. Salvi, J. L. Stein, Z. Filipi, and H. K. Fathy. “Development and model-based transparency analysis of an Internet-distributed hardware-in-the-loop simulation platform”. In: *Mechatronics* 21.1 (2011).
- [14] T. Ersal, M. Brudnak, A. Salvi, J. L. Stein, Z. Filipi, and H. K. Fathy. “Development of an Internet-distributed hardware-in-the-loop simulation platform for an automotive application”. In: *ASME 2009 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers. 2009, pp. 73–80.
- [15] M. Fischer, H. Sehammar, and G. Palmkvist. “Motion cueing for 3-, 6- and 8-degrees-of-freedom motion systems”. In: *Driving simulation conference*. Paris, France, 2010.
- [16] C. Fors, C. Ahlström, P. Sörner, J. Kovaceva, E. Hasselberg, M. Krantz, J.-F. Grönvall, K. Kircher, and A. Anund. *Camera-based sleepiness detection*. ViP publication 2011-6. Swedish National Road and Transport Research Institute, 2011.
- [17] P. Fritzson. *Principles of Object Oriented Modeling and Simulation with Modelica 3.3: A Cyber-Physical Approach*. Wiley-IEEE Press, 2014. ISBN: 978-1-118-85912-4.

-
- [18] J. Goodell, M. Compere, M. Simon, W. Smith, R. Wright, and M. Brudnak. “Robust Control Techniques for State Tracking in the Presence of Variable Time Delays”. In: *SAE Technical Paper*. SAE International, 2006.
 - [19] P. Hansson, A. Stenbeck, A. Kusachov, F. Bruzelius, and B. Augusto. “Prepositioning of driving simulator motion systems”. In: *International Journal of Vehicle Systems Modelling and Testing (IJVSMT)* 10.3 (2015).
 - [20] J. Jansson, J. Sandin, B. Augusto, and M. Fischer. “Design and performance of the VTI Sim IV”. In: *Driving simulation conference*. Paris, France, 2014.
 - [21] A. Kircher, K. Kircher, J. Törnros, A. Bolling, L. Nilsson, C. Patten, T. Malmström, and R. Ceci. *Mobile telephone simulator study*. VTI meddelande 969A. Swedish National Road and Transport Research Institute, 2004.
 - [22] G. Markkula, O. Benderius, K. Wolff, and M. Wahde. “Effects of experience and electronic stability control on low friction collision avoidance in a truck driving simulator”. In: *Accident Analysis and Prevention* 50 (2013), pp. 1266–1277.
 - [23] Modelica Association. *Modelica Language Specification 3.3*. www.modelica.org. 2012.
 - [24] B. Möller, K. L. Morse, M. Lightner, R. Little, and R. Lutz. “HLA Evolved – A Summary of Major Technical Improvements”. In: *Proceedings of the Fall 2008 Simulation Interoperability Workshop*. Orlando FL, 2008, pp. 15–19.
 - [25] *NADS Overview PDF*. <https://www.nads-sc.uiowa.edu/overview.php>. NADS, The University of Iowa. 2015.
 - [26] L. Nilsson. “Behavioural Research in an Advanced Driving Simulator – Experiences of the VTI System”. In: *Proceedings of the Human Factors and Ergonomics Society 37th Annual Meeting 1993*. 1993.
 - [27] S. Nordmark. *Driving simulators, trends and experiences*. VTI särtryck 204. Swedish National Road and Transport Research Institute, 1994.
 - [28] S. Nordmark. “The new Trygg Hansa truckdriving simulator: An advanced tool for research and training”. In: *Proceedings of the International Symposium on Advanced Vehicle Control 1992 (AVEC’92)*. 1992.
 - [29] S. Nordmark. *VTI Driving Simulator – Mathematical Model of a Four-wheeled Vehicle for Simulation in Real Time*. VTI rapport 267A. Swedish National Road and Transport Research Institute, 1984.

- [30] S. Nordmark, H. Jansson, G. Palmkvist, and H. Sehammar. “The new VTI Driving Simulator - Multi Purpose Moving Base with High Performance Linear Motion”. In: *Driving simulation conference*. Paris, France, 2004.
- [31] Open Source Modelica Consortium. *OpenModelica Users Guide version 1.9.0beta*. www.openmodelica.org. 2013.
- [32] C. Patten and S. Mårdh. *Förbifart Stockholm – utvärdering av ett gestaltungsförslag för tunnel*. VTI rapport 759. Swedish National Road and Transport Research Institute, 2012.
- [33] L. R. Petzold. “A description of DASSL: A differential/algebraic system solver”. In: *Scientific computing* 1 (1982), pp. 65–68.
- [34] M. Rothhämel. “Characterisation and Utilisation of Steering Feel in Heavy Trucks”. PhD thesis. Royal Institute of Technology, Sweden, 2013.
- [35] J. Törnros, L. Harms, and H. Alm. “The VTI driving simulator - Validation studies”. In: *DSC97 - Driving simulation conference*. Lyon, France, 1997.
- [36] J. Vinter. *SimArch Final report*. Tech. rep. Vinnova, 2010.

Papers

The papers associated with this thesis have been removed for copyright reasons. For more details about these see:

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-156537>

Appendix

List of Figures

1.1	The moving base driving simulators Sim3 and Sim4 at VTI.	2
1.2	Brake pressure distribution in the freight train. The left figure shows normal operation and the right figure shows the behavior when the changes in Listing 1.1 are applied.	5
3.1	A timeline that shows the process from when work within a project started until the resulting paper.	16
3.2	Chassis dynamometers set-up at Linköping University on the left and the driving simulator Sim III at the Swedish National Road and Transport Research Institute on the right.	17
3.3	Schematic view of the distributed driving simulator architecture using HLA.	18
3.4	Simulator setup using the moving base driving simulator Sim IV, a SCALEXIO hardware-in-the-loop simulator, the traffic simulation tool SUMO and a communication network between traffic actors called plexe-veins. The figure shows the setup that was used, where the traffic, network and driving simulator was located at VTI Sim IV. The HiL simulator at VICTA lab was connected to VTI using a VPN connection.	19
3.5	Validation run of a parameterized powertrain model for a front wheel drive Golf V. Parameterization only uses data collected from the chassis dynamometers and the car on-board diagnostics.	20
3.6	Collected driving data for two different routes in Linköping. Data contains sections with rural driving, lightly trafficked city driving, and motorway driving.	21

3.7	Mixed environment with interchangeable component interfaces in Modelica and hardware.	22
3.8	The left figure pictures a simple finite state machine used for a requirement. In the right figure, four requirements in Modelica are automatically verified for two different models and two different roads.	23
3.9	Simulator setup used during demonstration of the developed HLA-based simulation architecture in Gothenburg. SC, or Session Control, is controlling the simulation. ES, Environment Simulator, contains the road with its surrounding environment and vehicles. Sim IV, Desktop and VCC HiL are three different simulators. . . .	24
A.1	Schematic overview of the propulsion laboratory. The car is driven into the lab through the garage doors on the right and the dynamometers are then mounted to respective wheel hub. Road topography and vehicle parameters are then entered in the chassis dynamometer control PC. During the experiment the measurement and control PC in the control room forwards CAN messages from the chassis dynamometer to an external system via UDP. The external system then returns driver input to a human driver or, as in this case, a pedal robot.	40
A.2	The Volkswagen Passat that is used in the experiment mounted in the chassis dynamometer of the vehicle propulsion laboratory. . . .	43
A.3	The moving base driving simulator Sim III at VTI.	45
A.4	Pedal robot installed in the Volkswagen Passat test vehicle.	46
A.5	Position control of accelerator pedal with limits on integrator to avoid wind-up using Matlab Simulink.	47
A.6	Measured velocity from dynamometers rotational speed compared with calculated speed using (A.1) from measured dynamometers torque.	50
A.7	Longitudinal height profile of the used road segment.	51
A.8	Round trip time, RTT, between the propulsion lab at LiU and Sim III using the fibre optics link.	52
A.9	Step response tests made with the accelerator. The steps are from 0 (released) to 1 (maximally depressed) and from 0.25 to 0.75 of maximally depressed accelerator pedal. In the right figures five steps are shown on top of each other.	54
A.10	Step response tests made with the brake pedal. Steps are from minimum to maximum brake pressure (upper) and between 0.25 to 0.75 of maximum brake pressure (lower). Note the different control strategy for releasing the brake in the upper rightmost figure. . . .	54
A.11	Accelerator pedal position reference value compared with actuated position value for one of the test drivers.	55

A.12	Brake pedal pressure reference value compared with actuated pressure value for one of the test drivers.	56
A.13	Comparison of the Sim III vehicle speed with the chassis dynamometer vehicle speed for the test vehicle for the chosen K . . .	57
A.14	Response times from pedal position in Sim III to pedal robot position (upper) and from pedal robot position to test vehicle torque (lower).	58
B.1	Schematic view of the connection between SC, ES, DS and VS with interconnections.	64
B.2	State machine used in simulation entities.	66
B.3	Objects used within a federation.	68
B.4	Interactions which could be sent within the federation for controlling the simulation.	69
B.5	Objects used within the federation for communication between federates.	69
B.6	Interactions which could be sent within a simulator environment during a simulation as exchanging data.	70
B.7	Moving base simulator Sim III at VTI.	71
B.8	Moving base simulator Sim IV at VTI.	72
B.9	RTT measure within local network from server at VTI in Link ping and different simulators.	73
C.1	Moving base driving simulator “SimIV” at VTI.	81
C.2	Equipment at the VICTA Lab	82
C.3	Schematic view of the computer nodes and connections between them in the proposed simulation framework.	82
C.4	Functional view of the C-ITS simulation framework and the topology of VICTA Lab	84
C.5	Illustration of the Intersection Crossing Assist System scenario . .	85
C.6	Illustration of the Overtake Warning System scenario	86
C.7	Structures of the simulation framework tested in this work	87
D.1	A Golf V mounted to the chassis dynamometers at the Vehicle Propulsion Laboratory at LiU.	102
D.2	A dynamic measure of driveline torque $M_{w,dynamo} - M_{loss:drv}(i_g, \dot{\theta}_w)$ and wheel acceleration $\ddot{\theta}_w$	105
D.3	A spectral analysis of the wheel rotational acceleration signal. . .	106
D.4	Measured driveline losses, $M_{loss:drv}(g, \dot{\theta}_w)$, presented as a torque at the engine.	107
D.5	Measured accelerator pedal (left) and engine map (right) for a Golf V.	108

D.6	Measured accelerator pedal, throttle position and torque output compared to throttle and torque using the estimated accelerator pedal map and engine map for gear 2 (left) and gear 4 (right). . .	108
D.7	Measured throttle position depending on gear for different accelerator pedal positions.	110
D.8	A validation run for test case 3 going up to 5th gear and back down to 1st gear.	111
D.9	A validation run for test case 3 going up to 5th gear and back down to 1st gear.	112
D.10	The moving base simulator, Sim III, at VTI.	113
E.1	An overview of how the domains of operation and domains of validity are calculated and used to estimate vehicle dynamics model suitability for a driving task.	119
E.2	Schematic view of quality framework components used during a real-time simulation.	121
E.3	The chassis dynamometers and the test vehicle at the vehicle propulsion laboratory at Linköping University.	124
E.4	Data from trips done in two studies where mixed driving were performed. Each colour represent one test driver. Sections used to classify motorway driving are between 12-17 km in study 1 and 20-23 km in study 2.	125
E.5	Extracted engine torque and vehicle speed data from the two studies. The blue area marks operating points, and the ellipse marks the domain of operation for motorway driving.	126
E.6	Domain of operation for longitudinal and lateral acceleration from Study 1. The blue area marks operating points, and the ellipse marks the domain of operation for motorway driving.	126
E.7	Collected data from a driver driving the 130 km/h variant of the ARTEMIS Motorway Driving Cycle in the chassis dynamometers including driver input and vehicle speed.	127
E.8	Comparing two different discretization grids for collected motorway driving data with the domain of operation shown in red. Each blue dot represents one or more logged data points and a coarser grid is used in the right figure.	128
E.9	Estimated torque errors for each powertrain operational state (s_{tot}). States at which both clutch and acceleration pedals are pressed ($s_{acc}=1$ and $s_{clutch}=1$), are marked with a green circle. . .	129
E.10	Quality model state machine.	130
E.11	Domain of validity (black ellipse) for the developed powertrain model when running the 130 km/h variant of the ARTEMIS Motorway Driving Cycle. A green point illustrates a low error, a red point illustrates a large error, and intermediate errors are coloured between red and green.	131

E.12	Estimated errors when running the 130 km/h variant of the ARTEMIS Motorway Driving Cycle. Parts where the brake pedal has been used are removed.	132
E.13	A screen shot of the GUI for the simulator operator.	133
F.1	The VTI moving base Simulator III in Linköping.	143
F.2	Vehicle mounted at Vehicular Systems chassis dynamometer laboratory at LiU.	144
F.3	Pedal robot installed in the vehicle in the chassis dynamometer lab.	146
F.4	Comparison of engine rotational speed, blue curve, and vehicle wheel speeds multiplied with the gear ratio, red curve, at gear one.	150
F.5	Measured engine map from a car mounted in the chassis dynamometer lab.	151
F.6	A simulation of the complete chassis dynamometers lab model during a calm acceleration.	152
F.7	An overview of the components in the simulator environment. Green boxes picture hardware components and blue boxes picture Modelica model components. Red boxes picture components where either hardware or a Modelica model can be used.	153
F.8	Difference in acceleration when simulations are using Euler forward or DASSL as solvers during an acceleration maneuver.	154
G.1	The Golf V used for measuring powertrain data connected to chassis dynamometers at the vehicle propulsion laboratory at Linköping University.	164
G.2	Finite state machines used to model req1 above and req4 below. The requirement status corresponds to state circle format where a green line means NOT_VIOLATED , a red dashed line means VIOLATED and a blue dash dotted line means NOT_APPLICABLE	167
G.3	The complete model in OpenModelica with replaceable parts. The orange box marks the driver input which is modified depending on which data needs to be tested. The blue box is the model under test and is this is where the two different powertrain models are tested. The green box is the requirement model.	169
G.4	The powertrain model v1 and v2 during the 130 km/h variant of the Artemis Motorway Driving Cycle. The top four figures are the requirements one to four and the lower to figures show the difference in engine RPM and vehicle speed.	171
G.5	A close up of the first 55 seconds for models v1 and v2 during the 130 km/h variant of the Artemis Motorway Driving Cycle. Here req3 and req4 are shown.	172

List of Tables

2.1	A one tank system implemented in C++ and Modelica omitting routines for initialization and set and get input and/or output. . .	12
A.1	Vehicle model parameters used in the study.	44
A.2	Available output signals from the simulator rig.	44
A.3	Packet structure that was used in the RTT test between the two facilities.	52
A.4	Statistics for the RTT test between the facilities.	53
E.1	Parameter values used for the example domains presented in Figure E.5 and Figure E.6.	127
F.1	Used values for the vehicle model parameters.	145
F.2	Data sent from the chassis dynamometers.	145
F.3	Statistics from the connection between facilities.	147
F.4	Measured gear ratios from car mounted in the chassis dynamometer lab.	150
F.5	Time measurements from model simulation.	155
G.1	Textual requirement description.	166
G.2	Requirement status for the two powertrain models with two driving cycles.	172

Department of Computer and Information Science
Linköpings universitet

Dissertations

Linköping Studies in Science and Technology
Linköping Studies in Arts and Science

Linköping Studies in Statistics
Linköping Studies in Information Science

Linköping Studies in Science and Technology

- No 14 **Anders Haraldsson:** A Program Manipulation System Based on Partial Evaluation, 1977, ISBN 91-7372-144-1.
- No 17 **Bengt Magnhagen:** Probability Based Verification of Time Margins in Digital Designs, 1977, ISBN 91-7372-157-3.
- No 18 **Mats Cedwall:** Semantisk analys av process-beskrivningar i naturligt språk, 1977, ISBN 91-7372-168-9.
- No 22 **Jaak Urmi:** A Machine Independent LISP Compiler and its Implications for Ideal Hardware, 1978, ISBN 91-7372-188-3.
- No 33 **Tore Risch:** Compilation of Multiple File Queries in a Meta-Database System, 1978, ISBN 91-7372-232-4.
- No 51 **Erland Jungert:** Synthesizing Database Structures from a User Oriented Data Model, 1980, ISBN 91-7372-387-8.
- No 54 **Sture Hägglund:** Contributions to the Development of Methods and Tools for Interactive Design of Applications Software, 1980, ISBN 91-7372-404-1.
- No 55 **Pär Emanuelson:** Performance Enhancement in a Well-Structured Pattern Matcher through Partial Evaluation, 1980, ISBN 91-7372-403-3.
- No 58 **Bengt Johnsson, Bertil Andersson:** The Human-Computer Interface in Commercial Systems, 1981, ISBN 91-7372-414-9.
- No 69 **H. Jan Komorowski:** A Specification of an Abstract Prolog Machine and its Application to Partial Evaluation, 1981, ISBN 91-7372-479-3.
- No 71 **René Reboh:** Knowledge Engineering Techniques and Tools for Expert Systems, 1981, ISBN 91-7372-489-0.
- No 77 **Östen Oskarsson:** Mechanisms of Modifiability in large Software Systems, 1982, ISBN 91-7372-527-7.
- No 94 **Hans Lunell:** Code Generator Writing Systems, 1983, ISBN 91-7372-652-4.
- No 97 **Andrzej Lingas:** Advances in Minimum Weight Triangulation, 1983, ISBN 91-7372-660-5.
- No 109 **Peter Fritzson:** Towards a Distributed Programming Environment based on Incremental Compilation, 1984, ISBN 91-7372-801-2.
- No 111 **Erik Tengvald:** The Design of Expert Planning Systems. An Experimental Operations Planning System for Turning, 1984, ISBN 91-7372-805-5.
- No 155 **Christos Levcopoulos:** Heuristics for Minimum Decompositions of Polygons, 1987, ISBN 91-7870-133-3.
- No 165 **James W. Goodwin:** A Theory and System for Non-Monotonic Reasoning, 1987, ISBN 91-7870-183-X.
- No 170 **Zebo Peng:** A Formal Methodology for Automated Synthesis of VLSI Systems, 1987, ISBN 91-7870-225-9.
- No 174 **Johan Fagerström:** A Paradigm and System for Design of Distributed Systems, 1988, ISBN 91-7870-301-8.
- No 192 **Dimitar Driankov:** Towards a Many Valued Logic of Quantified Belief, 1988, ISBN 91-7870-374-3.
- No 213 **Lin Padgham:** Non-Monotonic Inheritance for an Object Oriented Knowledge Base, 1989, ISBN 91-7870-485-5.
- No 214 **Tony Larsson:** A Formal Hardware Description and Verification Method, 1989, ISBN 91-7870-517-7.
- No 221 **Michael Reinfrank:** Fundamentals and Logical Foundations of Truth Maintenance, 1989, ISBN 91-7870-546-0.
- No 239 **Jonas Löwgren:** Knowledge-Based Design Support and Discourse Management in User Interface Management Systems, 1991, ISBN 91-7870-720-X.
- No 244 **Henrik Eriksson:** Meta-Tool Support for Knowledge Acquisition, 1991, ISBN 91-7870-746-3.
- No 252 **Peter Eklund:** An Epistemic Approach to Interactive Design in Multiple Inheritance Hierarchies, 1991, ISBN 91-7870-784-6.
- No 258 **Patrick Doherty:** NML3 - A Non-Monotonic Formalism with Explicit Defaults, 1991, ISBN 91-7870-816-8.
- No 260 **Nahid Shahmehri:** Generalized Algorithmic Debugging, 1991, ISBN 91-7870-828-1.
- No 264 **Nils Dahlbäck:** Representation of Discourse-Cognitive and Computational Aspects, 1992, ISBN 91-7870-850-8.
- No 265 **Ulf Nilsson:** Abstract Interpretations and Abstract Machines: Contributions to a Methodology for the Implementation of Logic Programs, 1992, ISBN 91-7870-858-3.
- No 270 **Ralph Rönquist:** Theory and Practice of Tense-bound Object References, 1992, ISBN 91-7870-873-7.
- No 273 **Björn Fjellborg:** Pipeline Extraction for VLSI Data Path Synthesis, 1992, ISBN 91-7870-880-X.
- No 276 **Staffan Bonnier:** A Formal Basis for Horn Clause Logic with External Polymorphic Functions, 1992, ISBN 91-7870-896-6.
- No 277 **Kristian Sandahl:** Developing Knowledge Management Systems with an Active Expert Methodology, 1992, ISBN 91-7870-897-4.
- No 281 **Christer Bäckström:** Computational Complexity of Reasoning about Plans, 1992, ISBN 91-7870-979-2.
- No 292 **Mats Wirén:** Studies in Incremental Natural Language Analysis, 1992, ISBN 91-7871-027-8.
- No 297 **Mariam Kamkar:** Interprocedural Dynamic Slicing with Applications to Debugging and Testing, 1993, ISBN 91-7871-065-0.
- No 302 **Tingting Zhang:** A Study in Diagnosis Using Classification and Defaults, 1993, ISBN 91-7871-078-2.
- No 312 **Arne Jönsson:** Dialogue Management for Natural Language Interfaces - An Empirical Approach, 1993, ISBN 91-7871-110-X.
- No 338 **Simin Nadjm-Tehrani:** Reactive Systems in Physical Environments: Compositional Modelling and Framework for Verification, 1994, ISBN 91-7871-237-8.

- No 371 **Bengt Savén:** Business Models for Decision Support and Learning. A Study of Discrete-Event Manufacturing Simulation at Asea/ABB 1968-1993, 1995, ISBN 91-7871-494-X.
- No 375 **Ulf Söderman:** Conceptual Modelling of Mode Switching Physical Systems, 1995, ISBN 91-7871-516-4.
- No 383 **Andreas Kågedal:** Exploiting Groundness in Logic Programs, 1995, ISBN 91-7871-538-5.
- No 396 **George Fodor:** Ontological Control, Description, Identification and Recovery from Problematic Control Situations, 1995, ISBN 91-7871-603-9.
- No 413 **Mikael Pettersson:** Compiling Natural Semantics, 1995, ISBN 91-7871-641-1.
- No 414 **Xinli Gu:** RT Level Testability Improvement by Testability Analysis and Transformations, 1996, ISBN 91-7871-654-3.
- No 416 **Hua Shu:** Distributed Default Reasoning, 1996, ISBN 91-7871-665-9.
- No 429 **Jaime Villegas:** Simulation Supported Industrial Training from an Organisational Learning Perspective - Development and Evaluation of the SSIT Method, 1996, ISBN 91-7871-700-0.
- No 431 **Peter Jonsson:** Studies in Action Planning: Algorithms and Complexity, 1996, ISBN 91-7871-704-3.
- No 437 **Johan Boye:** Directional Types in Logic Programming, 1996, ISBN 91-7871-725-6.
- No 439 **Cecilia Sjöberg:** Activities, Voices and Arenas: Participatory Design in Practice, 1996, ISBN 91-7871-728-0.
- No 448 **Patrick Lambrix:** Part-Whole Reasoning in Description Logics, 1996, ISBN 91-7871-820-1.
- No 452 **Kjell Orsborn:** On Extensible and Object-Relational Database Technology for Finite Element Analysis Applications, 1996, ISBN 91-7871-827-9.
- No 459 **Olof Johansson:** Development Environments for Complex Product Models, 1996, ISBN 91-7871-855-4.
- No 461 **Lena Strömbäck:** User-Defined Constructions in Unification-Based Formalisms, 1997, ISBN 91-7871-857-0.
- No 462 **Lars Degerstedt:** Tabulation-based Logic Programming: A Multi-Level View of Query Answering, 1996, ISBN 91-7871-858-9.
- No 475 **Fredrik Nilsson:** Strategi och ekonomisk styrning - En studie av hur ekonomiska styrsystem utformas och används efter företagsförvärv, 1997, ISBN 91-7871-914-3.
- No 480 **Mikael Lindvall:** An Empirical Study of Requirements-Driven Impact Analysis in Object-Oriented Software Evolution, 1997, ISBN 91-7871-927-5.
- No 485 **Göran Forslund:** Opinion-Based Systems: The Cooperative Perspective on Knowledge-Based Decision Support, 1997, ISBN 91-7871-938-0.
- No 494 **Martin Sköld:** Active Database Management Systems for Monitoring and Control, 1997, ISBN 91-7219-002-7.
- No 495 **Hans Olsén:** Automatic Verification of Petri Nets in a CLP framework, 1997, ISBN 91-7219-011-6.
- No 498 **Thomas Drakengren:** Algorithms and Complexity for Temporal and Spatial Formalisms, 1997, ISBN 91-7219-019-1.
- No 502 **Jakob Axelsson:** Analysis and Synthesis of Heterogeneous Real-Time Systems, 1997, ISBN 91-7219-035-3.
- No 503 **Johan Ringström:** Compiler Generation for Data-Parallel Programming Languages from Two-Level Semantics Specifications, 1997, ISBN 91-7219-045-0.
- No 512 **Anna Moberg:** Närhet och distans - Studier av kommunikationsmönster i satellitkontor och flexibla kontor, 1997, ISBN 91-7219-119-8.
- No 520 **Mikael Ronström:** Design and Modelling of a Parallel Data Server for Telecom Applications, 1998, ISBN 91-7219-169-4.
- No 522 **Niclas Ohlsson:** Towards Effective Fault Prevention - An Empirical Study in Software Engineering, 1998, ISBN 91-7219-176-7.
- No 526 **Joachim Karlsson:** A Systematic Approach for Prioritizing Software Requirements, 1998, ISBN 91-7219-184-8.
- No 530 **Henrik Nilsson:** Declarative Debugging for Lazy Functional Languages, 1998, ISBN 91-7219-197-X.
- No 555 **Jonas Hallberg:** Timing Issues in High-Level Synthesis, 1998, ISBN 91-7219-369-7.
- No 561 **Ling Lin:** Management of 1-D Sequence Data - From Discrete to Continuous, 1999, ISBN 91-7219-402-2.
- No 563 **Eva I. Ragnemalm:** Student Modelling based on Collaborative Dialogue with a Learning Companion, 1999, ISBN 91-7219-412-X.
- No 567 **Jörgen Lindström:** Does Distance matter? On geographical dispersion in organisations, 1999, ISBN 91-7219-439-1.
- No 582 **Vanja Josifovski:** Design, Implementation and Evaluation of a Distributed Mediator System for Data Integration, 1999, ISBN 91-7219-482-0.
- No 589 **Rita Kovordányi:** Modeling and Simulating Inhibitory Mechanisms in Mental Image Reinterpretation - Towards Cooperative Human-Computer Creativity, 1999, ISBN 91-7219-506-1.
- No 592 **Mikael Ericsson:** Supporting the Use of Design Knowledge - An Assessment of Commenting Agents, 1999, ISBN 91-7219-532-0.
- No 593 **Lars Karlsson:** Actions, Interactions and Narratives, 1999, ISBN 91-7219-534-7.
- No 594 **C. G. Mikael Johansson:** Social and Organizational Aspects of Requirements Engineering Methods - A practice-oriented approach, 1999, ISBN 91-7219-541-X.
- No 595 **Jörgen Hansson:** Value-Driven Multi-Class Overload Management in Real-Time Database Systems, 1999, ISBN 91-7219-542-8.
- No 596 **Niklas Hallberg:** Incorporating User Values in the Design of Information Systems and Services in the Public Sector: A Methods Approach, 1999, ISBN 91-7219-543-6.
- No 597 **Vivian Vimarlund:** An Economic Perspective on the Analysis of Impacts of Information Technology: From Case Studies in Health-Care towards General Models and Theories, 1999, ISBN 91-7219-544-4.
- No 598 **Johan Jenvald:** Methods and Tools in Computer-Supported Taskforce Training, 1999, ISBN 91-7219-547-9.
- No 607 **Magnus Merkel:** Understanding and enhancing translation by parallel text processing, 1999, ISBN 91-7219-614-9.
- No 611 **Silvia Coradeschi:** Anchoring symbols to sensory data, 1999, ISBN 91-7219-623-8.
- No 613 **Man Lin:** Analysis and Synthesis of Reactive Systems: A Generic Layered Architecture Perspective, 1999, ISBN 91-7219-630-0.

- No 618 **Jimmy Tjäder:** Systemimplementering i praktiken - En studie av logiker i fyra projekt, 1999, ISBN 91-7219-657-2.
- No 627 **Vadim Engelson:** Tools for Design, Interactive Simulation, and Visualization of Object-Oriented Models in Scientific Computing, 2000, ISBN 91-7219-709-9.
- No 637 **Esa Falkenroth:** Database Technology for Control and Simulation, 2000, ISBN 91-7219-766-8.
- No 639 **Per-Arne Persson:** Bringing Power and Knowledge Together: Information Systems Design for Autonomy and Control in Command Work, 2000, ISBN 91-7219-796-X.
- No 660 **Erik Larsson:** An Integrated System-Level Design for Testability Methodology, 2000, ISBN 91-7219-890-7.
- No 688 **Marcus Bjärelund:** Model-based Execution Monitoring, 2001, ISBN 91-7373-016-5.
- No 689 **Joakim Gustafsson:** Extending Temporal Action Logic, 2001, ISBN 91-7373-017-3.
- No 720 **Carl-Johan Petri:** Organizational Information Provision - Managing Mandatory and Discretionary Use of Information Technology, 2001, ISBN 91-7373-126-9.
- No 724 **Paul Scerri:** Designing Agents for Systems with Adjustable Autonomy, 2001, ISBN 91-7373-207-9.
- No 725 **Tim Heyer:** Semantic Inspection of Software Artifacts: From Theory to Practice, 2001, ISBN 91-7373-208-7.
- No 726 **Pär Carlshamre:** A Usability Perspective on Requirements Engineering - From Methodology to Product Development, 2001, ISBN 91-7373-212-5.
- No 732 **Juha Takkinen:** From Information Management to Task Management in Electronic Mail, 2002, ISBN 91-7373-258-3.
- No 745 **Johan Åberg:** Live Help Systems: An Approach to Intelligent Help for Web Information Systems, 2002, ISBN 91-7373-311-3.
- No 746 **Rego Granlund:** Monitoring Distributed Teamwork Training, 2002, ISBN 91-7373-312-1.
- No 757 **Henrik André-Jönsson:** Indexing Strategies for Time Series Data, 2002, ISBN 91-7373-346-6.
- No 747 **Anneli Hagdahl:** Development of IT-supported Interorganisational Collaboration - A Case Study in the Swedish Public Sector, 2002, ISBN 91-7373-314-8.
- No 749 **Sofie Pilemalm:** Information Technology for Non-Profit Organisations - Extended Participatory Design of an Information System for Trade Union Shop Stewards, 2002, ISBN 91-7373-318-0.
- No 765 **Stefan Holmlid:** Adapting users: Towards a theory of use quality, 2002, ISBN 91-7373-397-0.
- No 771 **Magnus Morin:** Multimedia Representations of Distributed Tactical Operations, 2002, ISBN 91-7373-421-7.
- No 772 **Pawel Pietrzak:** A Type-Based Framework for Locating Errors in Constraint Logic Programs, 2002, ISBN 91-7373-422-5.
- No 758 **Erik Berglund:** Library Communication Among Programmers Worldwide, 2002, ISBN 91-7373-349-0.
- No 774 **Choong-ho Yi:** Modelling Object-Oriented Dynamic Systems Using a Logic-Based Framework, 2002, ISBN 91-7373-424-1.
- No 779 **Mathias Broxvall:** A Study in the Computational Complexity of Temporal Reasoning, 2002, ISBN 91-7373-440-3.
- No 793 **Asmus Pandikow:** A Generic Principle for Enabling Interoperability of Structured and Object-Oriented Analysis and Design Tools, 2002, ISBN 91-7373-479-9.
- No 785 **Lars Hult:** Publika Informationstjänster. En studie av den Internetbaserade encyklopedins bruksegenskaper, 2003, ISBN 91-7373-461-6.
- No 800 **Lars Taxén:** A Framework for the Coordination of Complex Systems' Development, 2003, ISBN 91-7373-604-X.
- No 808 **Klas Gäre:** Tre perspektiv på förväntningar och förändringar i samband med införande av informationssystem, 2003, ISBN 91-7373-618-X.
- No 821 **Mikael Kindborg:** Concurrent Comics - programming of social agents by children, 2003, ISBN 91-7373-651-1.
- No 823 **Christina Ölvingson:** On Development of Information Systems with GIS Functionality in Public Health Informatics: A Requirements Engineering Approach, 2003, ISBN 91-7373-656-2.
- No 828 **Tobias Ritzau:** Memory Efficient Hard Real-Time Garbage Collection, 2003, ISBN 91-7373-666-X.
- No 833 **Paul Pop:** Analysis and Synthesis of Communication-Intensive Heterogeneous Real-Time Systems, 2003, ISBN 91-7373-683-X.
- No 852 **Johan Moe:** Observing the Dynamic Behaviour of Large Distributed Systems to Improve Development and Testing - An Empirical Study in Software Engineering, 2003, ISBN 91-7373-779-8.
- No 867 **Erik Herzog:** An Approach to Systems Engineering Tool Data Representation and Exchange, 2004, ISBN 91-7373-929-4.
- No 872 **Aseel Berglund:** Augmenting the Remote Control: Studies in Complex Information Navigation for Digital TV, 2004, ISBN 91-7373-940-5.
- No 869 **Jo Skåmedal:** Telecommuting's Implications on Travel and Travel Patterns, 2004, ISBN 91-7373-935-9.
- No 870 **Linda Askenäs:** The Roles of IT - Studies of Organising when Implementing and Using Enterprise Systems, 2004, ISBN 91-7373-936-7.
- No 874 **Annika Flycht-Eriksson:** Design and Use of Ontologies in Information-Providing Dialogue Systems, 2004, ISBN 91-7373-947-2.
- No 873 **Peter Bunus:** Debugging Techniques for Equation-Based Languages, 2004, ISBN 91-7373-941-3.
- No 876 **Jonas Mellin:** Resource-Predictable and Efficient Monitoring of Events, 2004, ISBN 91-7373-956-1.
- No 883 **Magnus Bång:** Computing at the Speed of Paper: Ubiquitous Computing Environments for Healthcare Professionals, 2004, ISBN 91-7373-971-5.
- No 882 **Robert Eklund:** Disfluency in Swedish human-human and human-machine travel booking dialogues, 2004, ISBN 91-7373-966-9.
- No 887 **Anders Lindström:** English and other Foreign Linguistic Elements in Spoken Swedish. Studies of Productive Processes and their Modelling using Finite-State Tools, 2004, ISBN 91-7373-981-2.
- No 889 **Zhiping Wang:** Capacity-Constrained Production-inventory systems - Modelling and Analysis in both a traditional and an e-business context, 2004, ISBN 91-85295-08-6.
- No 893 **Pernilla Qvarfordt:** Eyes on Multimodal Interaction, 2004, ISBN 91-85295-30-2.
- No 910 **Magnus Kald:** In the Borderland between Strategy and Management Control - Theoretical Framework and Empirical Evidence, 2004, ISBN 91-85295-82-5.

- No 918 **Jonas Lundberg:** Shaping Electronic News: Genre Perspectives on Interaction Design, 2004, ISBN 91-85297-14-3.
- No 900 **Mattias Arvola:** Shades of use: The dynamics of interaction design for sociable use, 2004, ISBN 91-85295-42-6.
- No 920 **Luis Alejandro Cortés:** Verification and Scheduling Techniques for Real-Time Embedded Systems, 2004, ISBN 91-85297-21-6.
- No 929 **Diana Szentivanyi:** Performance Studies of Fault-Tolerant Middleware, 2005, ISBN 91-85297-58-5.
- No 933 **Mikael Cäker:** Management Accounting as Constructing and Opposing Customer Focus: Three Case Studies on Management Accounting and Customer Relations, 2005, ISBN 91-85297-64-X.
- No 937 **Jonas Kvarnström:** TALplanner and Other Extensions to Temporal Action Logic, 2005, ISBN 91-85297-75-5.
- No 938 **Bourhane Kadmiry:** Fuzzy Gain-Scheduled Visual Servoing for Unmanned Helicopter, 2005, ISBN 91-85297-76-3.
- No 945 **Gert Jervan:** Hybrid Built-In Self-Test and Test Generation Techniques for Digital Systems, 2005, ISBN 91-85297-97-6.
- No 946 **Anders Arpteg:** Intelligent Semi-Structured Information Extraction, 2005, ISBN 91-85297-98-4.
- No 947 **Ola Angelsmark:** Constructing Algorithms for Constraint Satisfaction and Related Problems - Methods and Applications, 2005, ISBN 91-85297-99-2.
- No 963 **Calin Curescu:** Utility-based Optimisation of Resource Allocation for Wireless Networks, 2005, ISBN 91-85457-07-8.
- No 972 **Björn Johansson:** Joint Control in Dynamic Situations, 2005, ISBN 91-85457-31-0.
- No 974 **Dan Lawesson:** An Approach to Diagnosability Analysis for Interacting Finite State Systems, 2005, ISBN 91-85457-39-6.
- No 979 **Claudiu Duma:** Security and Trust Mechanisms for Groups in Distributed Services, 2005, ISBN 91-85457-54-X.
- No 983 **Sorin Manolache:** Analysis and Optimisation of Real-Time Systems with Stochastic Behaviour, 2005, ISBN 91-85457-60-4.
- No 986 **Yuxiao Zhao:** Standards-Based Application Integration for Business-to-Business Communications, 2005, ISBN 91-85457-66-3.
- No 1004 **Patrik Haslum:** Admissible Heuristics for Automated Planning, 2006, ISBN 91-85497-28-2.
- No 1005 **Aleksandra Tešanovic:** Developing Reusable and Reconfigurable Real-Time Software using Aspects and Components, 2006, ISBN 91-85497-29-0.
- No 1008 **David Dinka:** Role, Identity and Work: Extending the design and development agenda, 2006, ISBN 91-85497-42-8.
- No 1009 **Iakov Nakhimovski:** Contributions to the Modeling and Simulation of Mechanical Systems with Detailed Contact Analysis, 2006, ISBN 91-85497-43-X.
- No 1013 **Wilhelm Dahllöf:** Exact Algorithms for Exact Satisfiability Problems, 2006, ISBN 91-85523-97-6.
- No 1016 **Levon Saldamli:** PDEModelica - A High-Level Language for Modeling with Partial Differential Equations, 2006, ISBN 91-85523-84-4.
- No 1017 **Daniel Karlsson:** Verification of Component-based Embedded System Designs, 2006, ISBN 91-85523-79-8.
- No 1018 **Ioan Chisalita:** Communication and Networking Techniques for Traffic Safety Systems, 2006, ISBN 91-85523-77-1.
- No 1019 **Tarja Susi:** The Puzzle of Social Activity - The Significance of Tools in Cognition and Cooperation, 2006, ISBN 91-85523-71-2.
- No 1021 **Andrzej Bednarski:** Integrated Optimal Code Generation for Digital Signal Processors, 2006, ISBN 91-85523-69-0.
- No 1022 **Peter Aronsson:** Automatic Parallelization of Equation-Based Simulation Programs, 2006, ISBN 91-85523-68-2.
- No 1030 **Robert Nilsson:** A Mutation-based Framework for Automated Testing of Timeliness, 2006, ISBN 91-85523-35-6.
- No 1034 **Jon Edvardsson:** Techniques for Automatic Generation of Tests from Programs and Specifications, 2006, ISBN 91-85523-31-3.
- No 1035 **Vaida Jakoniene:** Integration of Biological Data, 2006, ISBN 91-85523-28-3.
- No 1045 **Genevieve Gorrell:** Generalized Hebbian Algorithms for Dimensionality Reduction in Natural Language Processing, 2006, ISBN 91-85643-88-2.
- No 1051 **Yu-Hsing Huang:** Having a New Pair of Glasses - Applying Systemic Accident Models on Road Safety, 2006, ISBN 91-85643-64-5.
- No 1054 **Åsa Hedenskog:** Perceive those things which cannot be seen - A Cognitive Systems Engineering perspective on requirements management, 2006, ISBN 91-85643-57-2.
- No 1061 **Cécile Åberg:** An Evaluation Platform for Semantic Web Technology, 2007, ISBN 91-85643-31-9.
- No 1073 **Mats Grindal:** Handling Combinatorial Explosion in Software Testing, 2007, ISBN 978-91-85715-74-9.
- No 1075 **Almut Herzog:** Usable Security Policies for Runtime Environments, 2007, ISBN 978-91-85715-65-7.
- No 1079 **Magnus Wahlström:** Algorithms, measures, and upper bounds for Satisfiability and related problems, 2007, ISBN 978-91-85715-55-8.
- No 1083 **Jesper Andersson:** Dynamic Software Architectures, 2007, ISBN 978-91-85715-46-6.
- No 1086 **Ulf Johansson:** Obtaining Accurate and Comprehensive Data Mining Models - An Evolutionary Approach, 2007, ISBN 978-91-85715-34-3.
- No 1089 **Traian Pop:** Analysis and Optimisation of Distributed Embedded Systems with Heterogeneous Scheduling Policies, 2007, ISBN 978-91-85715-27-5.
- No 1091 **Gustav Nordh:** Complexity Dichotomies for CSP-related Problems, 2007, ISBN 978-91-85715-20-6.
- No 1106 **Per Ola Kristensson:** Discrete and Continuous Shape Writing for Text Entry and Control, 2007, ISBN 978-91-85831-77-7.
- No 1110 **He Tan:** Aligning Biomedical Ontologies, 2007, ISBN 978-91-85831-56-2.
- No 1112 **Jessica Lindblom:** Minding the body - Interacting socially through embodied action, 2007, ISBN 978-91-85831-48-7.
- No 1113 **Pontus Wärnestål:** Dialogue Behavior Management in Conversational Recommender Systems, 2007, ISBN 978-91-85831-47-0.
- No 1120 **Thomas Gustafsson:** Management of Real-Time Data Consistency and Transient Overloads in Embedded Systems, 2007, ISBN 978-91-85831-33-3.

- No 1127 **Alexandru Andrei**: Energy Efficient and Predictable Design of Real-time Embedded Systems, 2007, ISBN 978-91-85831-06-7.
- No 1139 **Per Wikberg**: Eliciting Knowledge from Experts in Modeling of Complex Systems: Managing Variation and Interactions, 2007, ISBN 978-91-85895-66-3.
- No 1143 **Mehdi Amirijoo**: QoS Control of Real-Time Data Services under Uncertain Workload, 2007, ISBN 978-91-85895-49-6.
- No 1150 **Sanny Syberfeldt**: Optimistic Replication with Forward Conflict Resolution in Distributed Real-Time Databases, 2007, ISBN 978-91-85895-27-4.
- No 1155 **Beatrice Alenljung**: Envisioning a Future Decision Support System for Requirements Engineering - A Holistic and Human-centred Perspective, 2008, ISBN 978-91-85895-11-3.
- No 1156 **Artur Wilk**: Types for XML with Application to Xcerpt, 2008, ISBN 978-91-85895-08-3.
- No 1183 **Adrian Pop**: Integrated Model-Driven Development Environments for Equation-Based Object-Oriented Languages, 2008, ISBN 978-91-7393-895-2.
- No 1185 **Jörgen Skågeby**: Gifting Technologies - Ethnographic Studies of End-users and Social Media Sharing, 2008, ISBN 978-91-7393-892-1.
- No 1187 **Imad-Eldin Ali Abugessaisa**: Analytical tools and information-sharing methods supporting road safety organizations, 2008, ISBN 978-91-7393-887-7.
- No 1204 **H. Joe Steinhauer**: A Representation Scheme for Description and Reconstruction of Object Configurations Based on Qualitative Relations, 2008, ISBN 978-91-7393-823-5.
- No 1222 **Anders Larsson**: Test Optimization for Core-based System-on-Chip, 2008, ISBN 978-91-7393-768-9.
- No 1238 **Andreas Borg**: Processes and Models for Capacity Requirements in Telecommunication Systems, 2009, ISBN 978-91-7393-700-9.
- No 1240 **Fredrik Heintz**: DyKnow: A Stream-Based Knowledge Processing Middleware Framework, 2009, ISBN 978-91-7393-696-5.
- No 1241 **Birgitta Lindström**: Testability of Dynamic Real-Time Systems, 2009, ISBN 978-91-7393-695-8.
- No 1244 **Eva Blomqvist**: Semi-automatic Ontology Construction based on Patterns, 2009, ISBN 978-91-7393-683-5.
- No 1249 **Rogier Woltjer**: Functional Modeling of Constraint Management in Aviation Safety and Command and Control, 2009, ISBN 978-91-7393-659-0.
- No 1260 **Gianpaolo Conte**: Vision-Based Localization and Guidance for Unmanned Aerial Vehicles, 2009, ISBN 978-91-7393-603-3.
- No 1262 **AnnMarie Ericsson**: Enabling Tool Support for Formal Analysis of ECA Rules, 2009, ISBN 978-91-7393-598-2.
- No 1266 **Jiri Trnka**: Exploring Tactical Command and Control: A Role-Playing Simulation Approach, 2009, ISBN 978-91-7393-571-5.
- No 1268 **Bahlol Rahimi**: Supporting Collaborative Work through ICT - How End-users Think of and Adopt Integrated Health Information Systems, 2009, ISBN 978-91-7393-550-0.
- No 1274 **Fredrik Kuivinen**: Algorithms and Hardness Results for Some Valued CSPs, 2009, ISBN 978-91-7393-525-8.
- No 1281 **Gunnar Mathiason**: Virtual Full Replication for Scalable Distributed Real-Time Databases, 2009, ISBN 978-91-7393-503-6.
- No 1290 **Viacheslav Izosimov**: Scheduling and Optimization of Fault-Tolerant Distributed Embedded Systems, 2009, ISBN 978-91-7393-482-4.
- No 1294 **Johan Thapper**: Aspects of a Constraint Optimisation Problem, 2010, ISBN 978-91-7393-464-0.
- No 1306 **Susanna Nilsson**: Augmentation in the Wild: User Centered Development and Evaluation of Augmented Reality Applications, 2010, ISBN 978-91-7393-416-9.
- No 1313 **Christer Thörn**: On the Quality of Feature Models, 2010, ISBN 978-91-7393-394-0.
- No 1321 **Zhiyuan He**: Temperature Aware and Defect-Probability Driven Test Scheduling for System-on-Chip, 2010, ISBN 978-91-7393-378-0.
- No 1333 **David Broman**: Meta-Languages and Semantics for Equation-Based Modeling and Simulation, 2010, ISBN 978-91-7393-335-3.
- No 1337 **Alexander Siemers**: Contributions to Modelling and Visualisation of Multibody Systems Simulations with Detailed Contact Analysis, 2010, ISBN 978-91-7393-317-9.
- No 1354 **Mikael Asplund**: Disconnected Discoveries: Availability Studies in Partitioned Networks, 2010, ISBN 978-91-7393-278-3.
- No 1359 **Jana Rambusch**: Mind Games Extended: Understanding Gameplay as Situated Activity, 2010, ISBN 978-91-7393-252-3.
- No 1373 **Sonia Sangari**: Head Movement Correlates to Focus Assignment in Swedish, 2011, ISBN 978-91-7393-154-0.
- No 1374 **Jan-Erik Källhammer**: Using False Alarms when Developing Automotive Active Safety Systems, 2011, ISBN 978-91-7393-153-3.
- No 1375 **Mattias Eriksson**: Integrated Code Generation, 2011, ISBN 978-91-7393-147-2.
- No 1381 **Ola Leifler**: Affordances and Constraints of Intelligent Decision Support for Military Command and Control - Three Case Studies of Support Systems, 2011, ISBN 978-91-7393-133-5.
- No 1386 **Soheil Samii**: Quality-Driven Synthesis and Optimization of Embedded Control Systems, 2011, ISBN 978-91-7393-102-1.
- No 1419 **Erik Kuiper**: Geographic Routing in Intermittently-connected Mobile Ad Hoc Networks: Algorithms and Performance Models, 2012, ISBN 978-91-7519-981-8.
- No 1451 **Sara Stymne**: Text Harmonization Strategies for Phrase-Based Statistical Machine Translation, 2012, ISBN 978-91-7519-887-3.
- No 1455 **Alberto Montebelli**: Modeling the Role of Energy Management in Embodied Cognition, 2012, ISBN 978-91-7519-882-8.
- No 1465 **Mohammad Saifullah**: Biologically-Based Interactive Neural Network Models for Visual Attention and Object Recognition, 2012, ISBN 978-91-7519-838-5.
- No 1490 **Tomas Bengtsson**: Testing and Logic Optimization Techniques for Systems on Chip, 2012, ISBN 978-91-7519-742-5.
- No 1481 **David Byers**: Improving Software Security by Preventing Known Vulnerabilities, 2012, ISBN 978-91-7519-784-5.
- No 1496 **Tommy Färnqvist**: Exploiting Structure in CSP-related Problems, 2013, ISBN 978-91-7519-711-1.

- No 1503 **John Wilander:** Contributions to Specification, Implementation, and Execution of Secure Software, 2013, ISBN 978-91-7519-681-7.
- No 1506 **Magnus Ingmarsson:** Creating and Enabling the Useful Service Discovery Experience, 2013, ISBN 978-91-7519-662-6.
- No 1547 **Wladimir Schamai:** Model-Based Verification of Dynamic System Behavior against Requirements: Method, Language, and Tool, 2013, ISBN 978-91-7519-505-6.
- No 1551 **Henrik Svensson:** Simulations, 2013, ISBN 978-91-7519-491-2.
- No 1559 **Sergiu Rafiliu:** Stability of Adaptive Distributed Real-Time Systems with Dynamic Resource Management, 2013, ISBN 978-91-7519-471-4.
- No 1581 **Usman Dastgeer:** Performance-aware Component Composition for GPU-based Systems, 2014, ISBN 978-91-7519-383-0.
- No 1602 **Cai Li:** Reinforcement Learning of Locomotion based on Central Pattern Generators, 2014, ISBN 978-91-7519-313-7.
- No 1652 **Roland Samlaus:** An Integrated Development Environment with Enhanced Domain-Specific Interactive Model Validation, 2015, ISBN 978-91-7519-090-7.
- No 1663 **Hannes Uppman:** On Some Combinatorial Optimization Problems: Algorithms and Complexity, 2015, ISBN 978-91-7519-072-3.
- No 1664 **Martin Sjölund:** Tools and Methods for Analysis, Debugging, and Performance Improvement of Equation-Based Models, 2015, ISBN 978-91-7519-071-6.
- No 1666 **Kristian Stavåker:** Contributions to Simulation of Modelica Models on Data-Parallel Multi-Core Architectures, 2015, ISBN 978-91-7519-068-6.
- No 1680 **Adrian Lifa:** Hardware/Software Codesign of Embedded Systems with Reconfigurable and Heterogeneous Platforms, 2015, ISBN 978-91-7519-040-2.
- No 1685 **Bogdan Tanasa:** Timing Analysis of Distributed Embedded Systems with Stochastic Workload and Reliability Constraints, 2015, ISBN 978-91-7519-022-8.
- No 1691 **Håkan Warnquist:** Troubleshooting Trucks – Automated Planning and Diagnosis, 2015, ISBN 978-91-7685-993-3.
- No 1702 **Nima Aghaee:** Thermal Issues in Testing of Advanced Systems on Chip, 2015, ISBN 978-91-7685-949-0.
- No 1715 **Maria Vasilevskaya:** Security in Embedded Systems: A Model-Based Approach with Risk Metrics, 2015, ISBN 978-91-7685-917-9.
- No 1729 **Ke Jiang:** Security-Driven Design of Real-Time Embedded System, 2016, ISBN 978-91-7685-884-4.
- No 1733 **Victor Lagerkvist:** Strong Partial Clones and the Complexity of Constraint Satisfaction Problems: Limitations and Applications, 2016, ISBN 978-91-7685-856-1.
- No 1734 **Chandan Roy:** An Informed System Development Approach to Tropical Cyclone Track and Intensity Forecasting, 2016, ISBN 978-91-7685-854-7.
- No 1746 **Amir Aminifar:** Analysis, Design, and Optimization of Embedded Control Systems, 2016, ISBN 978-91-7685-826-4.
- No 1747 **Ekhiotz Vergara:** Energy Modelling and Fairness for Efficient Mobile Communication, 2016, ISBN 978-91-7685-822-6.
- No 1748 **Dag Sonntag:** Chain Graphs – Interpretations, Expressiveness and Learning Algorithms, 2016, ISBN 978-91-7685-818-9.
- No 1768 **Anna Vapen:** Web Authentication using Third-Parties in Untrusted Environments, 2016, ISBN 978-91-7685-753-3.
- No 1778 **Magnus Jandinger:** On a Need to Know Basis: A Conceptual and Methodological Framework for Modelling and Analysis of Information Demand in an Enterprise Context, 2016, ISBN 978-91-7685-713-7.
- No 1798 **Rahul Hiran:** Collaborative Network Security: Targeting Wide-area Routing and Edge-network Attacks, 2016, ISBN 978-91-7685-662-8.
- No 1813 **Nicolas Melot:** Algorithms and Framework for Energy Efficient Parallel Stream Computing on Many-Core Architectures, 2016, ISBN 978-91-7685-623-9.
- No 1823 **Amy Rankin:** Making Sense of Adaptations: Resilience in High-Risk Work, 2017, ISBN 978-91-7685-596-6.
- No 1831 **Lisa Malmberg:** Building Design Capability in the Public Sector: Expanding the Horizons of Development, 2017, ISBN 978-91-7685-585-0.
- No 1851 **Marcus Bendtsen:** Gated Bayesian Networks, 2017, ISBN 978-91-7685-525-6.
- No 1852 **Zlatan Dragisic:** Completion of Ontologies and Ontology Networks, 2017, ISBN 978-91-7685-522-5.
- No 1854 **Meysam Aghighi:** Computational Complexity of some Optimization Problems in Planning, 2017, ISBN 978-91-7685-519-5.
- No 1863 **Simon Ståhlberg:** Methods for Detecting Unsolvable Planning Instances using Variable Projection, 2017, ISBN 978-91-7685-498-3.
- No 1879 **Karl Hammar:** Content Ontology Design Patterns: Qualities, Methods, and Tools, 2017, ISBN 978-91-7685-454-9.
- No 1887 **Ivan Ukhov:** System-Level Analysis and Design under Uncertainty, 2017, ISBN 978-91-7685-426-6.
- No 1891 **Valentina Ivanova:** Fostering User Involvement in Ontology Alignment and Alignment Evaluation, 2017, ISBN 978-91-7685-403-7.
- No 1902 **Vengatanathan Krishnamoorthi:** Efficient HTTP-based Adaptive Streaming of Linear and Interactive Videos, 2018, ISBN 978-91-7685-371-9.
- No 1903 **Lu Li:** Programming Abstractions and Optimization Techniques for GPU-based Heterogeneous Systems, 2018, ISBN 978-91-7685-370-2.
- No 1913 **Jonas Rybing:** Studying Simulations with Distributed Cognition, 2018, ISBN 978-91-7685-348-1.
- No 1936 **Leif Jonsson:** Machine Learning-Based Bug Handling in Large-Scale Software Development, 2018, ISBN 978-91-7685-306-1.
- No 1964 **Arian Maghazeh:** System-Level Design of GPU-Based Embedded Systems, 2018, ISBN 978-91-7685-175-3.

- No 1967 **Mahder Gebremedhin:** Automatic and Explicit Parallelization Approaches for Equation Based Mathematical Modeling and Simulation, 2019, ISBN 978-91-7685-163-0.
- No 1984 **Anders Andersson:** Distributed Moving Base Driving Simulators – Technology, Performance, and Requirements, 2019, ISBN 978-91-7685-090-9.

Linköping Studies in Arts and Science

- No 504 **Ing-Marie Jonsson:** Social and Emotional Characteristics of Speech-based In-Vehicle Information Systems: Impact on Attitude and Driving Behaviour, 2009, ISBN 978-91-7393-478-7.
- No 586 **Fabian Segelström:** Stakeholder Engagement for Service Design: How service designers identify and communicate insights, 2013, ISBN 978-91-7519-554-4.
- No 618 **Johan Blomkvist:** Representing Future Situations of Service: Prototyping in Service Design, 2014, ISBN 978-91-7519-343-4.
- No 620 **Marcus Mast:** Human-Robot Interaction for Semi-Autonomous Assistive Robots, 2014, ISBN 978-91-7519-319-9.
- No 677 **Peter Berggren:** Assessing Shared Strategic Understanding, 2016, ISBN 978-91-7685-786-1.
- No 695 **Mattias Forsblad:** Distributed cognition in home environments: The prospective memory and cognitive practices of older adults, 2016, ISBN 978-91-7685-686-4.

Linköping Studies in Statistics

- No 9 **Davood Shahsavani:** Computer Experiments Designed to Explore and Approximate Complex Deterministic Models, 2008, ISBN 978-91-7393-976-8.
- No 10 **Karl Wahlin:** Roadmap for Trend Detection and Assessment of Data Quality, 2008, ISBN 978-91-7393-792-4.
- No 11 **Oleg Sysoev:** Monotonic regression for large multivariate datasets, 2010, ISBN 978-91-7393-412-1.
- No 13 **Agné Burauskaite-Harju:** Characterizing Temporal Change and Inter-Site Correlations in Daily and Sub-daily Precipitation Extremes, 2011, ISBN 978-91-7393-110-6.
- No 14 **Måns Magnusson:** Scalable and Efficient Probabilistic Topic Model Inference for Textual Data, 2018, ISBN 978-91-7685-288-0.

Linköping Studies in Information Science

- No 1 **Karin Axelsson:** Metodisk systemstrukturerings- att skapa samstämmighet mellan informationssystemarkitektur och verksamhet, 1998, ISBN 9172-19-296-8.
- No 2 **Stefan Cronholm:** Metodverktyg och användbarhet - en studie av datorstödd metodbaserad systemutveckling, 1998, ISBN 9172-19-299-2.
- No 3 **Anders Avdic:** Användare och utvecklare - om anveckling med kalkylprogram, 1999, ISBN 91-7219-606-8.
- No 4 **Owen Eriksson:** Kommunikationskvalitet hos informationssystem och affärsprocesser, 2000, ISBN 91-7219-811-7.
- No 5 **Mikael Lind:** Från system till process - kriterier för processbestämning vid verksamhetsanalys, 2001, ISBN 91-7373-067-X.
- No 6 **Ulf Melin:** Koordination och informationssystem i företag och nätverk, 2002, ISBN 91-7373-278-8.

- No 7 **Pär J. Ågerfalk:** Information Systems Actability - Understanding Information Technology as a Tool for Business Action and Communication, 2003, ISBN 91-7373-628-7.
- No 8 **Ulf Seigerroth:** Att förstå och förändra systemutvecklingsverksamheter - en taxonomi för metautveckling, 2003, ISBN 91-7373-736-4.
- No 9 **Karin Hedström:** Spår av datoriseringens värden - Effekter av IT i äldreomsorg, 2004, ISBN 91-7373-963-4.
- No 10 **Ewa Braf:** Knowledge Demanded for Action - Studies on Knowledge Mediation in Organisations, 2004, ISBN 91-85295-47-7.
- No 11 **Fredrik Karlsson:** Method Configuration method and computerized tool support, 2005, ISBN 91-85297-48-8.
- No 12 **Malin Nordström:** Styrbar systemförvaltning - Att organisera systemförvaltningsverksamhet med hjälp av effektiva förvaltningsobjekt, 2005, ISBN 91-85297-60-7.
- No 13 **Stefan Holgersson:** Yrke: POLIS - Yrkeskunskap, motivation, IT-system och andra förutsättningar för polisarbete, 2005, ISBN 91-85299-43-X.
- No 14 **Benneth Christiansson, Marie-Therese Christiansson:** Mötet mellan process och komponent - mot ett ramverk för en verksamhetsnära kravspecifikation vid anskaffning av komponentbaserade informationssystem, 2006, ISBN 91-85643-22-X.

FACULTY OF SCIENCE AND ENGINEERING

Linköping Studies in Science and Technology, Dissertations, No. 1984, 2019
Department of Computer and Information Science

Linköping University
SE-581 83 Linköping, Sweden

www.liu.se