Mälardalen University Press Dissertations

No. 52

ENGINEERING AUTOMOTIVE ELECTRONIC SYSTEMS: DECISION SUPPORT FOR SUCCESSFUL INTEGRATION

Joakim Fröberg

2007

MÄLARDALEN UNIVERSITY
SWEDEN

Department of Computer Science and Electronics

ISSN 1651-4238
ISBN 978-91-85485-65-9
Printed by Arkitektkopia, Västerås, Sweden

Abstract

Development of a modern vehicle involves integration of components from several organizations. Many components are mechatronic which means that they include mechanical parts together with electronics that actively enhance some property of the component. The electronic system is increasingly important to product behaviour as vehicle functions control and coordinates more of these mechatronic subsystems. As the operational functionality increase, the amount of built-in functions for lifecycle support such as production tests and diagnostics also increase. Integrated electronic components assists in all these goals of the system. Achieving a successful integration where the components fit is accomplished in the development phases preceding the integration. Problems in integration often lead to severe delays close to the start of vehicle production.

This thesis presents results on the subject of integration of automotive electronic systems. Our studies aim at providing knowledge on how to integrate automotive electronic systems successfully in a setting where vehicles are developed based on existing platforms. We focus on early phases of automotive electronic system development and in particular on the decisions taken in integration of electronic subsystems. The contribution is the presented support for making decisions to successfully integrate electronic systems for modern vehicles. The contribution includes an overview of driving factors of automotive electronics system design, a validated set of success practices for the integration of electronic components, and the proposal and demonstration of a decision model. The influential factors and the validated set of practices stems from case studies of products and projects while the proposed decision model is a result of combining two general models for architecture analysis and decision making, the Architecture Tradeoff Analysis Method, ATAM and the Analytic Hierarchy Process, AHP.

We demonstrate that choices in strategy and design preceding integration are central to achieve a successful integration. Our studies show that problems arise from omitted strategy decisions and we provide a checklist for decision making in the areas; functionality, platform, integration design, and assigning responsibilities. We provide a recommendation that we validate in a multiple cases study where fulfilment of recommendations is demonstrated to affect project success in integration projects. The potential gain for OEMs using our results lies in achieving more solid foundations for design decisions. Designers and managers could potentially find central decisions on integration strategy early that, if omitted, could cause delays. Thus, applying the result could avoid pitfalls and enable successful integration projects.

Acknowledgements

I would like to thank my supervisors Christer Norström and Kristian Sandström. Endless enthusiasm and constructive advice have made my work a joy.

I would also like to thank Nils-Erik Bånkestad, whose insights has guided my studies and who somehow always finds time to give invaluable input and advice to my work. This work would have been something entirely different without Nils-Erik.

Several people have helped me with ideas, hard work, discussions, and enthusiasm. I would like to thank especially Jakob Axelsson, Mikael Åkerholm, Anders Möller, Björn Villing, Mikael Nolin, Stig Larsson, Peter Wallin, Jack Samuelsson.

It is difficult to describe the inspiring atmosphere I have been working in during my Phd studies. The great and fun environment comes from the great and fun people that I work with. I would like to thank all my friends and colleagues at Volvo and IDE who makes my working atmosphere such a rewarding one.

Finally, I would like to thank you Marie. You inspire me always!

Joakim Fröberg, Västerås, November 2007

“One day Alice came to a fork in the road and saw a Cheshire cat in a tree. Which road do I take? she asked. Where do you want to go? was his response. I don't know, Alice answered. Then, said the cat, it doesn't matter.”

Alice's Adventures in Wonderland

Lewis Carroll - 1832-1898

List of Included Papers

A. Business Situation Reflected in Automotive Electronic Architectures: Analysis of Four Commercial Cases, Joakim Fröberg, Kristian Sandström, Christer Norström, 2nd International ICSE workshop on Software Engineering for Automotive Systems, St. Louis, May, 2005

   This paper presents a case study with data from four automotive development organizations and provides analysis on the relation between business situation and the resulting electronic system architecture. I contributed by leading the study while all co-authors were involved in data collection, validating, and analysis.

   There are other results from this study presented in Papers 1, 9, and 11 in the list of related papers. Paper 11 is focused on the networking technology aspects of the study. The authors of this paper are listed in alphabetical order but I was the first author. Paper 9 is a conference paper based on the results. Paper 1 is a monographic licentiate thesis that includes the results from the case study but also sections on automotive industry and trends. Paper A is a short version with the most important results of all these papers.

B. Key Factors for Achieving Project Success in Integration of Automotive Mechatronics, Joakim Fröberg, Mikael Åkerholm, Kristian Sandström, Christer Norström, Journal of Innovations in Systems and Software Engineering, vol 11334 2007/3/16, p15, Springer, April, 2007

   I contributed by leading the studies and the analysis. This result is based on two case studies and the first was presented in paper 4. My contribution in paper 4 was to lead the study and the conclusion and analysis was made in cooperation with Mikael Åkerholm.

C. Making Decisions in Integration of Automotive Software and Electronics: A Method Based on ATAM and AHP, Peter Wallin, Joakim Fröberg, Jakob Axelsson, 4th International ICSE workshop on Software Engineering for Automotive Systems, Minneapolis, USA, May, 2007

I am not the first author of this paper, and I contributed with 50% of the work. This paper is based on the paper "Towards Quality Assessment in Integration of Automotive Software and Electronics: An ATAM approach" in which I was the first author. Basically I contributed with the idea of using the ATAM like evaluation to support design decisions and the theoretical example. Peter Wallin contributed with all information on AHP and CPC and together we devised the combined model and demonstrated its use.

Other results

Papers 5, 7, and 10 in the list of related papers present a study on industrial requirements for software component technology for automotive products. I contributed by, together with Anders Möller and Mikael Åkerholm, devising the study and performing data collection, mostly interviews. In addition, I aided in the analysis.

Paper 6 presents a study of software architecture in several companies in the embedded domain. Goran Mustapic led the study and I contributed with data collection and analysis in the two cases that were performed in automotive companies.

Paper 2 presents a process improvement with proposed tool support for developing reusable software components for the automotive domain. I contributed by aiding in the analysis.

List of Related Papers

Licentiate thesis

1. Engineering of Vehicle Electronic Systems: Requirements Reflected in Architecture, Joakim Fröberg, Licentiate Thesis, Mälardalen University Press, April, 2004

Conferences and workshops

2. A Model for Reuse and Optimization of Embedded Software Components, Mikael Åkerholm, Joakim Fröberg, Kristian Sandström, Ivica Crnkovic, 29th International Conference on Information technology Interface, (ITI 2007), IEEE, Cavtat, Croatia, June, 2007
3. Towards Quality Assessment in Integration of Automotive Software and Electronics: An ATAM approach, Joakim Fröberg, Peter Wallin, Jakob Axelsson, Proceedings of the 6th Conference on Software Engineering and Practice in Sweden, Umeå University, Umeå, Sweden, October, 2006
4. Integration of Electronic Components in Heavy Vehicles: A Study of Integration in Three Cases, Joakim Fröberg, Mikael Åkerholm, Proceedings from Systems Engineering/Test and Evaluation Conference, Melbourne, 25-27 September 2006, Melbourne, September, 2006
5. Industrial Grading of Quality Requirements for Automotive Software Component Technologies, Anders Möller, Mikael Åkerholm, Joakim Fröberg, Mikael Nolin, Embedded Real-Time Systems Implementation Workshop in conjunction with the 26th IEEE International Real-Time Systems Symposium, 2005 Miami, USA, December, 2005
6. Real World Influences on Software Architecture - Interviews with Industrial Experts, Goran Mustapic, Anders Wall, Christer Norström, Ivica Crnkovic, Kristian Sandström, Joakim Fröberg,

Johan Andersson, IEEE Working Conferance on Software Architectures, Oslo, Norway, IEEE, Oslo, Editor(s):IEEE, June, 2004

7. Industrial Requirements on Component Technologies for Embedded Systems, Anders Möller, Joakim Fröberg, Mikael Nolin, International Symposium on Component-based Software Engineering (CBSE7), Springer Verlag, Edinburgh, Scotland, May, 2004
8. What are the needs for components in vehicular systems? - An industrial perspective, Anders Möller, Joakim Fröberg, Mikael Nolin, Real-Time in Sweden (RTiS), MRTC, Västerås, Sweden, August, 2003
9. Correlating Bussines Needs and Network Architectures in Automotive Applications - a Comparative Case Study, Joakim Fröberg, Kristian Sandström, Christer Norström, Hans Hansson, Jakob Axelsson, Björn Villing (external), Proceedings of the 5th IFAC International Conference on Fieldbus Systems and their Applications (FET), p 219-228, IFAC, Aveiro, Portugal, July, 2003
10. What are the needs for components in vehicular systems? - An industrial perspective -, Anders Möller, Joakim Fröberg, Mikael Nolin, Proceedings of the WiP Session of the 15th Euromicro Conference on Real-Time Systems, p 45 - 48, Porto, Portugal, July, 2003.
11. A Comparative Case Study of Distributed Network Architectures for Different Automotive Applications, Jakob Axelsson, Joakim Fröberg, Hans Hansson, Christer Norström, Kristian Sandström, Björn Villing (external), The Industrial Information Technology Handbook, p 57-1 to 57-20, CRC Press, ISBN: 0-8493-1985-4, Editor(s): Richard Zurawski, January, 2005

Contents

CHAPTER 1. INTRODUCTION 1
1.1 Research questions 6
1.2 Thesis outline 7
CHAPTER 2. CONTRIBUTION AND MAIN FINDINGS............. 9
2.1 Results 9
2.1.1 Study A - Business drivers and architectures . 10
2.1.2 Study B - Integration project success 12
2.1.3 Study C – Decision support model 18
2.2 Applying the results.. 19
CHAPTER 3. RESEARCH METHOD....................................... 21
3.1 Study A – Business drivers and architectures 21
3.2 Study B – Integration project success........................... 24
3.3 Study C – Decision support model................................ 27
CHAPTER 4. RELATED WORK 31
4.1 Study A – Business drivers and architectures 31
4.2 Study B – Integration project success........................... 32
4.3 Study C – Decision support model................................ 35
4.4 Summary 36
CHAPTER 5. DISCUSSION AND FUTURE WORK ...... ERROR! BOOKMARK NOT DEFINED.
CHAPTER 6. CONCLUSION 37
CHAPTER 7. REFERENCES 41

Chapter 1.
Introduction

Development of a modern vehicle is performed by joining components developed by several organizations; both internal and external to the vehicle manufacturer, referred to as the Original Equipment Manufacturer, OEM. Automotive OEMs desire both the benefits in cost and functionality by using specialized suppliers. Some of the ingredient components are available at the outset of a new vehicle development project while others may need to be developed. Today, vehicle components are often mechatronic, meaning that they include mechanical parts together with electronics that actively enhance some property of the component.

The embedded electronic system in a vehicle is central to achieving a successful product and vehicle development is increasingly focused on electronic systems [1]. Electronics is involved and assist in achieving multiple goals in a modern vehicle. Vehicle properties such as comfort or handling, as well as optimized energy or performance, can be accomplished by distributed electronic functions coordinating vehicle subsystems. In addition to enhanced vehicle usage, life cycle aspects of the vehicle need to be satisfied by the electronic system, e.g., system self-diagnosis and built in functional tests for production.

The goal when designing a complete vehicle is to achieve a product that is optimal for its life cycle. Figure 1. shows the life cycle of a vehicle product involving development, manufacturing, use at customer, maintenance including service and repairs, and disposal. The product is to exhibit properties to support or enable these phases. Throughout the vehicle's life cycle, numerous stakeholders require different things in order to handle their phase of the vehicles life cycle. The arrows in the figure show a range of desired properties that stem from different stakeholder requirements. The property requirements originate from an internal document at Volvo Cars. The range and diversity of requirements illustrates the basic notion of complexity in design of automotive products. In order to make design decisions, we must understand what properties are required and which design choices achieves them. Each component should be designed to assist in achieving these vehicle system properties. Thus, each component is simultaneously involved in many goals.

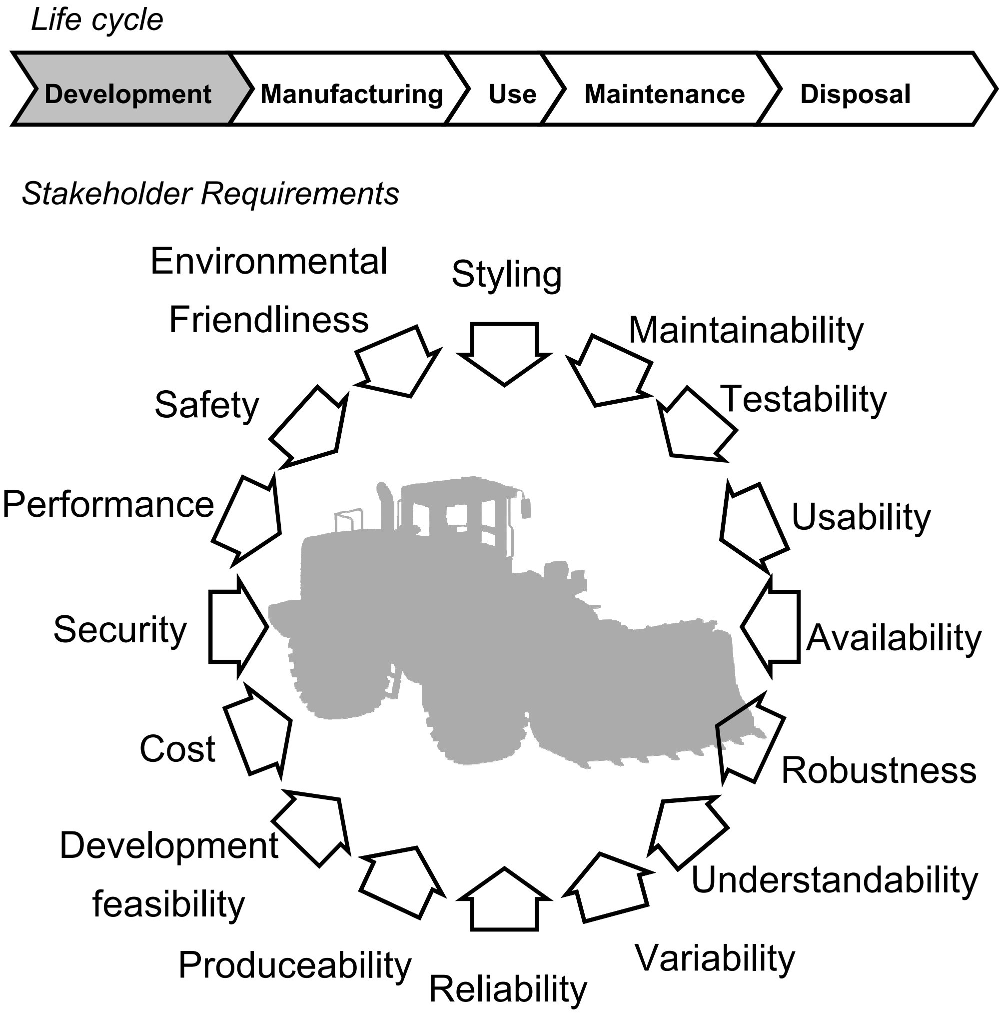

Figure 1. Life cycle and stakeholders requirements

The architecture of a system is generally considered to define its properties and architecture design is the activity that is aimed at solving complex sets of requirements. The IEEE defines architecture as “the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its evolution and design” [12]. This definition does not identify what choices are architectural and we consider the fundamental organization and principles as being the important design decisions. Which these are in a given system can be identified by architecture analysis methods such as the Architecture Tradeoff Analysis Method, ATAM [6].

For an OEM some of these important design decisions are fixed. An OEM uses a vehicle platform as a basis for developing a new vehicle model. A vehicle platform includes system architecture, components, technology, process and tools that are common to, and reused in several vehicle models. In the same way, an electronic platform is defined by a series of design decisions and choices in electronic system architecture, electronic components, process, tools, etc. The main reasons for using a platform are to achieve goals in quality and cost through component sharing.

In this thesis, we refer to the act of joining the electronic components as *integration*. This integration is done when components are becoming available in the later stages of development, but automotive systems typically include numerous platform components that may have been available early. Therefore, we also refer to integration as *integrating* components with a platform.

Integration of components, like pieces of a puzzle, however, is straightforward if the components fit perfectly. “Click”, Integration. This click is difficult to achieve because of the complexity involved. The system is to exhibit many functions and properties, and can operate in many different modes. Thus, the contact surface or interface of the components is multi dimensional. A component should be integrated in a way that all the system level goals are met. Systems engineering tasks such as planning, choice of concepts and strategy is what precedes the integration phase. On the way to a perfect “click” integration, we need to decide how a component is to function and decide on technical solutions to interact within the system and best meet the numerous goals. Axelsson argues in [2] that the key issue in handling increasing automotive systems complexity is systems integration, which calls for increasing systems engineering capabilities.

Failures in deciding the strategy and concepts of electronic integration can have severe impact on the development schedule for a complete vehicle. Electronic systems are integrated late in a vehicle project and a failure in integration possibly results in a late change of schedule. Electronic systems are integrated late simply because they are not ready before mechanical parts are finalized and controlling electronics can then be finally verified. Integration is not just an activity that will be prolonged if a failure is met. Instead, a failure indicates a flawed design and can force setbacks to earlier phases. If the component does not fit, the mismatch should be fixed and this could cause rethinking of central concepts.

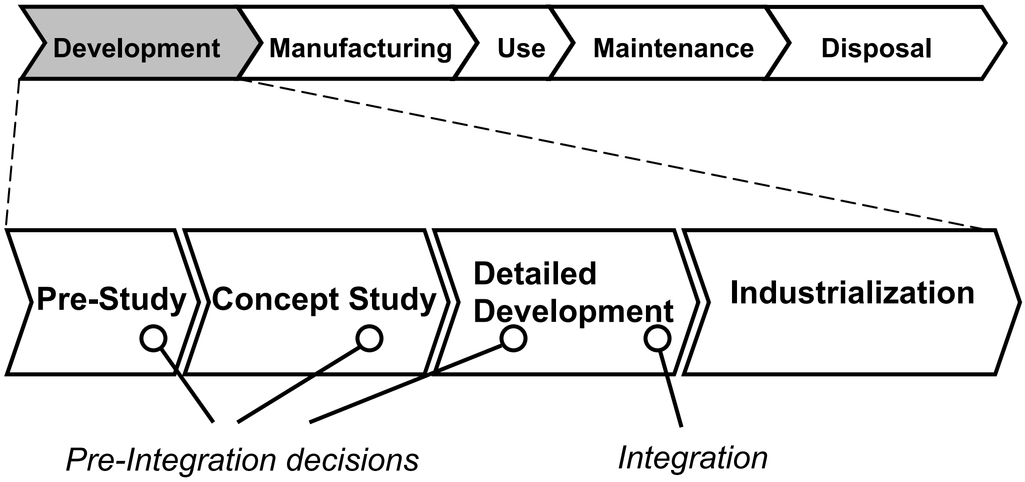

Figure 2. Integration in a product development process

In this description, we would like to point out two important aspects of electronic integration.

1. Integration is preceded by decisions on how the system components should interact, decisions that depend on the many system goals.
2. Achieving successful integration is important because a failure can cause delay close to start of production, a delay that is associated with large costs and reduced revenues.

Figure 2. shows a generic development process based on the stage gate model [11] and shows that integration of components take place late in the detailed development phase. The detailed development phase is where the components of the product are implemented and verified. Only then can the integration take place. Preceding the detailed development phase is the pre-study and the concept study phase. The pre-study involves eliciting requirements and the concept study involves defining the product architecture, which is the division into subsystems with defined responsibilities. When the functionality of the subsystems such as hydraulics, electronics, and electrics has been decided, the detailed development phase can start with implementing subsystems as planned and then to integrate them into a product.

We have, in this thesis focused on the decisions preceding the integration and aim at providing support for successful integration. We attempt to provide support for making decisions on integration; which decisions are important and how to judge which choice among candidates is the best.

Example

In order to illustrate we use an example. We describe a realistic example that involves design drivers, life cycle, and design and integration decisions and illustrates how they relate to the process of electronic integration.

Service is an important aspect of the life cycle of a vehicle. In order for service shops to be cost effective, the diagnosis of a vehicle must be performed efficiently using only one service tool, i.e., a laptop computer running a single software tool. Using several tools would cause longer service stops for all vehicles with a resulting increase in time and cost. This is one of the driving requirements, which is listed among others in the pre-study. In the pre-study phase, the overall requirements are extracted from analyzing the market and a feature list and business case is produced. This phase may unveil that customers desire some improved property of, e.g., the brakes, such as longer maintenance interval, better performance or less noise. The concept phase may then show several suppliers that have a readily developed mechatronic brake systems that meet or can be made to meet the criteria. The concept phase involves evaluating and choosing a vehicle architecture and part of this would be to decide which braking system should be chosen. In order to evaluate candidates, criteria are set up and one such criterion is the feasibility of electronic integration. One of the issues in electronic integration would then be to decide on how to meet the aforementioned requirement of having diagnostic support in an OEM specific tool. The chosen brake system must, among other things, be made to signal its self-diagnosis information to the OEM service tool via the electronic system of the vehicle in a way that conforms to electronic platform standards. This and all other aspects of interaction with the vehicle system are decided as part of the integration strategy.

Diagnostic strategy decisions, on the other hand, are part of the platform and a product project cannot change them. Instead, platform changes are made in a separate and longer life cycle for the platform product.

Industry anecdote

The issue of electronic integration is visible in industry magazines. One problem related to integration was reported in 2004 [4]. The OEM Mercedes was forced to recall 680 000 cars to fix a defect in an integrated brake system from the supplier Bosch by applying a software patch. Three years earlier, there was a failure of an integrated command system that made use of the navigator, audio system, and car phone. “It is difficult to integrate these gadgets into a vehicle’s infrastructure, said Stephan Wolfried, who is Mercedes-Benz’s vice president for electrical and electronics and chassis development”. This case shows both that there are problems related to integration and that shortcomings can have severe consequences.

1.1 Research questions

Problem

The problem with integration of electronic components in automotive electronics systems is essentially that failures cause project delay and that this is found out late. Failures in integration can lead to large extra costs in development and possibly delayed production. Some of the underlying reasons for failing in integration can be related to; 1, understanding requirements, 2, executing integration projects, and 3, making design decisions.

1. Decisions on design and integration can turn out wrong if the requirements are not known. Handling requirements from numerous stakeholders and choosing a design that best meet them is a challenge.

2. In addition, the time of integration can reveal problems if concepts and strategies for integration are not decided. These are worked out during the development project and there seems to be many factors for a complex system to consider.

3. Design decisions are difficult to make when numerous and conflicting requirements apply. In addition, it is difficult to choose between alternatives when requirements are imprecise.

Objectives

The objective of this study is to improve knowledge on how to integrate electronic systems in automotive products. Finding methods and models to enable a more solid foundation for design decisions would potentially enable a more successful integration and thus affect quality, development costs, and project risks.

Research questions

How to design and integrate electronic systems are major questions and this research is set up to provide answers to some of these. In short, this research aims at providing knowledge on how to integrate automotive electronic systems successfully in a setting of existing platforms.

In order to do this we have set up our studies according to three research questions. First, we want to explore the OEM business situation; the requirements and the design space in terms of what architectures are employed in contemporary vehicles.

Q1. What design drivers do exist in the business situation of OEMs and how do they affect design in practical cases?

Our second question is aimed at providing results on how to perform electronic integration projects involving mechatronic components. We aim at providing a recommendation on factors that should not be omitted in projects including mechatronic integration.

Q2. What practices and decisions lead to success in integration projects?

A structured design that takes into account all the possibly conflicting and inexact requirements is sought. An important aspect when proposing a new design method is to recognize an automotive system complexity and explicitly address the practice of design by integration.

Q3. How can decision making in integration be supported by structured methods?

By answering these questions, we aim to provide support for decision making for electronic integration in early phases of development work in the domain of automotive electronic systems. First, we aim to explore the domain of designing automotive electronic architectures and provide explanations on the relation between business situation and system architecture. For integration projects, we aim to identify factors that are critical to success and provide guidance for how to enable structured reasoning on design choices where the requirements are complex and imprecise.

1.2 Thesis outline

Part 1 of this thesis presents an overview of the research where the results are compiled. Chapter 1. presents an introduction to design of automotive electronic systems and integration of electronic components. The problem of integration is outlined and our research questions are defined. Chapter 2. presents the contributions and main findings from our studies. In Chapter 3. we describe the research methods we have used together with reasoning on validity. Chapter 4. presents related work and **Error! Reference source not found.**includes discussion and conclusion of the thesis.

Part 2 of this thesis includes papers A-C:

Paper A

“Business Situation Reflected in Automotive Electronic Architectures: Analysis of Four Commercial Cases”

Paper B

"Key Factors for Achieving Project Success in Integration of Automotive Mechatronics"

Paper C

"Making Decisions in Integration of Automotive Software and Electronics: A Method Based on ATAM and AHP"

Chapter 2.
Contribution and main findings

2.1 Results

In this section, we list the results from our studies. Figure 3. shows where our result are aimed to support decisions.

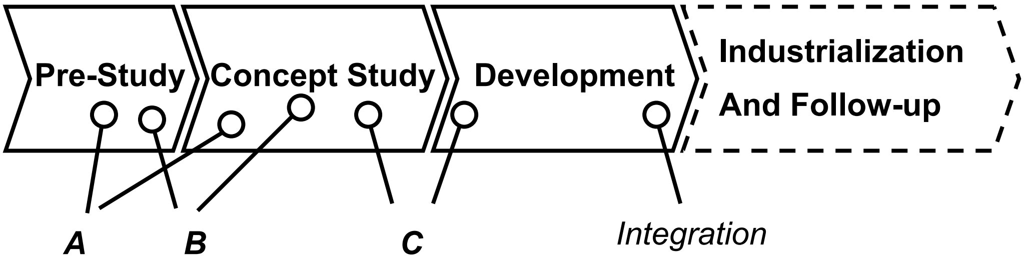

Figure 3. Integration in a product development process

In study A, we investigate the relation between requirements and design choices in architecture design. The requirements from the business situation are elicited in the pre-study phase and the design of system architecture is addressed in the concept phase. Study B shows recommendations with checklists for achieving successful integration. The decisions preceding the time of integration are demonstrated as critical. The study indicates primarily that decisions in the concept phase are critical, but also some decisions are identified that have to do with requirements in the pre-study phase. Study C is also primarily intended at supporting decisions in the concept phase, but more detailed design decisions could be targeted as well. In summary the results are aimed at supporting decisions in early phases of development including concept and strategy choices.

2.1.1 Study A - Business drivers and architectures

Our first research question (Q1) is addressed by performing a multiple case study.

Q1. What design drivers do exist in the business situation of OEMs and how do they affect the electronic architecture in practical cases?

Studying design drivers for automotive systems, we have presented case studies of the business situation, context, and architecture choices in four companies developing busses, construction equipment, trucks, and passenger cars. In the study, we have identified challenges with respect to functionality, cost, standards, and architecture for development of vehicles. Based on these case studies with different business and functionality demands, we have provided analysis of the design principles used for the communication architectures in these domains. Despite a common base of similar vehicle functionality the resulting network architectures used by the three organizations are quite different. The study shows four functionally rather similar products with computer controlled power train, body functions, and instrument. In the light of the business situation, we explain the solutions and why design principles are pursued. The reason for this diversity becomes apparent when looking at different business and product characteristics and their affect on the network architecture. An important lesson from this is that one should be very careful to uncritically apply technical solutions from one industry in another, even when they are as closely related as the applications described in this work. Understanding the requirements from the business case is the key to choosing architectural solutions.

The results in study A illustrate some of the design drivers that exist in the business situation of automotive OEMs. Table 1 in Paper A shows some key figures of business context including product volume, number of products, number of vehicle platforms, the size of the development organization, and the OEM market share. Table 2 shows measurements of key parameters in the business requirements for each of the four cases. This includes; the number of product variants, the focus on commonality, the need for hardware optimization, the need for system openness, the demand for customer adaptations, demands for infotainment, and demand for telematics. Table 3 shows key figures on the employed electronics system architecture; the amount of physical configurations per product, the amount of network information, the standards used on the network application level, the number of network technologies, the number of internally developed nodes, and the number of external node suppliers.

For the four studies cases we provide an analysis of how these design drivers affect design and choice of architecture. Some of the key drivers identified in the study are;

- The production volume drives demands for optimization, which means an increased inclination for the OEM to consider specialized solutions to reduce hardware cost. The willingness to reduce variable cost (product hardware cost) at the expense of fixed cost (investments and development costs) increases with the product volume. The OEM can use development resources to tailor designs and use many variants to meet optimization requirements.
- Commonality is not only desired as a result of high product volume, but also due to the potential savings in life-cycle operations with factories and service shops handling a minimum of different physical articles. For software, a high number articles or software versions would not affect costs in the same way, but would put strain on the working process and configuration management.
- The methods used to integrate supplier components and functionality differs among the four organizations. Having requirements on openness where other organization are to add superstructures to the vehicle electronic system stands out as forcing the use of communication standards. Having no demand for openness enables the use of proprietary protocols and a more precise specification of supplier component interaction.
- The size of the market is an important factor that enables suppliers to target produce large volumes and thereby provide components at lower cost than would OEM internal development.

Our analysis of the relation between key parameters in business context, business requirements, and resulting architectural solutions has shown that technical consideration alone cannot explain the choices in architecture. Parameters that affect architectural choices are product volume, market size, and requirements for openness and customer adaptation.

The trends in automotive industry indicates that the areas of model based development tools, standardized software architecture, improved network technologies are areas which could potentially target some of the requirements presented in our study.

2.1.2 Study B - Integration project success

Research question number two (Q2) is addressed by study B and the result is presented in paper B.

Q2. What key factors lead to success in integration projects?

The main contribution of study B is the validated recommendations, each including a set of checkpoints that defines recommendation fulfillment. We present a multiple case study on integration of automotive mechatronic components and based on the findings, we identify that the root causes of problems in integration are largely related to decisions omitted in electronic strategy.

Checklists to counteract reported problems

Our interviews with specialists reveal a number of decisions that, if omitted, reportedly affect project success in integration projects. We list these decisions in Table 1.

Group	***Grouping***	***Decision***
R1 - Functionality		Timing
		Operation
		Fault behavior
	Diagnostics	Data reporting
		Software upgrade
		Calibration
R2 - Platform		System modes
		Functional principles
		Protocols
		Proprietary extensions
		Tools
R3 - Integration -SW	Platform	Compiler
		Execution model
		Platform functionality
	Resource consumption	Memory
		CPU
		Bus
R3 - Integration -HW		Physical interface
	Environmental	EMC
		Moist
		Dust
		Vibration
R4 - Responsibilities		Maintenance
	Ownership	Service
	Ownership	Upgrades
		Electronics

Table 1. Decision checkpoints.

These decisions are also shown in Paper B, in figures 2 through 5. Each decision checkpoint in column 3 of Table 1 represents a strategy that should be decided in order to comply with that recommendation. The checkpoints are grouped together into four main areas of concern corresponding to our first four recommendations R1- R4.

R1 - Functionality

We demonstrate that decisions on functionality, if omitted, lead to unsuccessful projects. The reason for such a failure seems to be the difficulty in understanding the range of functionality support where support for service and production functionality seems especially under prioritized. Especially, the study shows that much of the focus prior to choosing component is on the operational functionality of the component while diagnostic functions and system interaction issues are omitted. Examples are system degradation behavior, fault signaling, and calibration, all of which often constitute a major part of the electronic system. Another typical problem reported was that the detailed technical issues of protocols, interfaces, and tools were wrongly estimated to be adaptable.

R2 - Platform

In addition, we demonstrate that omitting to address platform constraints when deciding on integration issues lead to unsuccessful projects. Failure in knowing the constraints of the electronic platform seems to arise because of complexity and difficulties in estimating impact.

Each checkpoint, thus, involves knowing one or more constraints and deciding to adhere to it. The critical decisions to take according to the study results are listed in the group R2 in Table 1. The checkpoints are divided into constraints related to the infrastructure of the system and constraints related to choices in technology and standards. The infrastructure of an automotive electronic platform does include some mechanism to support different system modes and also it may involve functional principles or inherent system philosophies. The platform have explicit system modes such as safe mode, key modes, and perhaps other operational modes, and the component must provide functionality to support this. It must be known what system modes in the platform that is relevant to the component to be integrated to fulfill the checkpoint. The platform can also contain other principles of operation. System design principles can include paradigms such as time triggered software execution or bus communication, or a client server architecture in software. In the category of technology and standard restrictions, communication protocols are mentioned together with company proprietary extensions. Standards and OEM extensions stipulate syntax and semantics of messages on a communication bus and therefore limit the design space for integration. Interview data also show that tool dependencies have been unclear and supposedly caused problems in integration.

R3 - Integration

Failure to address decisions on the integration solution is also shown to be a defining factor on integration success. If omitted, seemingly minor issues such as a conflicting bus message id, has later proved to be problematic to change.

Here, there are two basic choices in integration strategy as shown in Table 1. Either the strategy is to integrate an ECU on communication busses in the system (integration HW), or to integrate software functionality into an existing ECU (integration SW). The checklist for actions is different in the two strategies. Basically, in order to select the optimal component, we suggest evaluating both strategies and compare the effort needed given the wanted functionality. However, if there are reasons why the strategy cannot be freely chosen, the checklist can be applied for only the selected strategy, i.e., hardware or software integration.

For hardware integration, the checklist includes decisions to make for physical interface and environmental requirements on physical parts. For instance, for a given functionality, the ECU may need to be connected to several networks and this should be explicitly decided and feasibility should be assessed. Also, there are decisions to make for environmental requirements. These are likely specified by standards and there may be different areas of the vehicle that implies different physical roughness. These decisions should be explicitly stated and agreed upon with suppliers.

For software integration the focus is largely different. A software component can be integrated by deciding and specifying the software platform interface for the intended ECU host. Decisions should be made on compiler dependencies, execution model, and software platform services. Also, the resource consumption of a software component should be decided because there are limited system resources.

R4 - Responsibilities

The last area of affecting decisions we identify is the area of stakeholder involvement and assigning of responsibilities. The investigated cases show incompleteness in responsibilities as one likely reason for delay and increased project cost. There were several departments within the OEM that initiated projects involving electronics. Also the electronic system spans most of the vehicle subsystems and it was not always decided what role was to be responsible for each electronic subsystem. Reportedly, roles in service, maintenance and electronics were not fully decided. Also ownership of designs was mentioned as a potential pitfall for the project outcome.

Project success compared to fulfillment of checklists

In order to measure the success of each project, we have collected data on how the project was planned at the time of choosing the component. We use three measures and compare the initial plan with the actual outcome. We look at the projected time of completion, the projected product cost for the component, and the projected development cost.

We have used a Likert scale with numbers 1 to 5 for estimates of each checkpoint from table 1 and calculated an average for each R. We have also used a Likert scale of 1 to 5 for the measures on project success and calculated an average. The results are shown in Table 2 below.

The details of each recommendation and the legend for measurements are presented in paper B.

	R1	*R2*	*R3*	*R4*	*Average*	*Project Success*
Case #1	4.0	3.7	4.5	3.7	4.0	4.0
Case #2	4.2	4.3	4.5	3.7	4.2	4.3
Case #3	2.2	2.0	1.7	2.0	2.0	1.3
Case #4	3.2	3.0	3.7	3.7	3.4	3.0
Case #5	3.2	4.3	4.7	4.3	4.2	3.7

Table 2. Decision checkpoints.

We see that there is a correlation between fulfillment of the recommendations and the achieved project success. Although the numbers are just indicative, the trend can be seen that the recommendations do affect project outcome.

Recommendations R1-R6

We present six recommendations; the first four including detailed checklists for decision-making. Also, the first four are validated by measurements while the last two are results of our analysis.

1. All the functionality of the component should be decided prior to designing the integration solution; this includes diagnosis, production, and service functions.

2. Know the design constraints imposed by the platform prior to designing integration solution, e.g., global systems modes, communication protocols, and all constraining paradigms.
3. The integration solutions should be investigated and a strategy chosen prior to choosing component; this should include investigation of environmental requirements, and resource consumption.
4. All stakeholders should be involved and the responsibilities should be assigned for the activities of the subsystem life cycle.
5. Review decisions on integration and check that delivered components match decisions as soon as possible, to detect misconceptions early.
6. Be aware that integration projects characterized by a technically tight integration, safety criticality, close relation to core vehicle behavior, or inexperienced suppliers are high-risk projects.

Our analysis shows two more recommendations (5-6) that potentially would have affected the outcome in the studied cases. Avoiding mismatch between what is believed to be decided and what an involved supplier delivers could have been accomplished by follow-ups during the projects. Trying to estimate the difficulty in an integration project, we note that projects involving safety related functions, large impact on product behavior, technically advanced integration, or inexperienced suppliers could mean a higher risk of project failure.

A second result presented in paper B is the defining characteristics to identify a high-risk project. We provide a set of observable project properties and demonstrate how they indicate increased project risk.

Explanation of recommendations

The respondents did express problems and we concluded that they were mostly concerned with decisions. Consequently, we grouped all reported lacking decisions in areas that labeled and described that set in a good way. However, there is an element of refinement in this step as we add a notion of when the decisions are to be made. Some assumptions were made on how the flow of development progresses. Here we explicitly explain our reasoning of time and the relation to a development model. Figure 4. shows a graph describing the sequence of events in which the recommendation is defined.

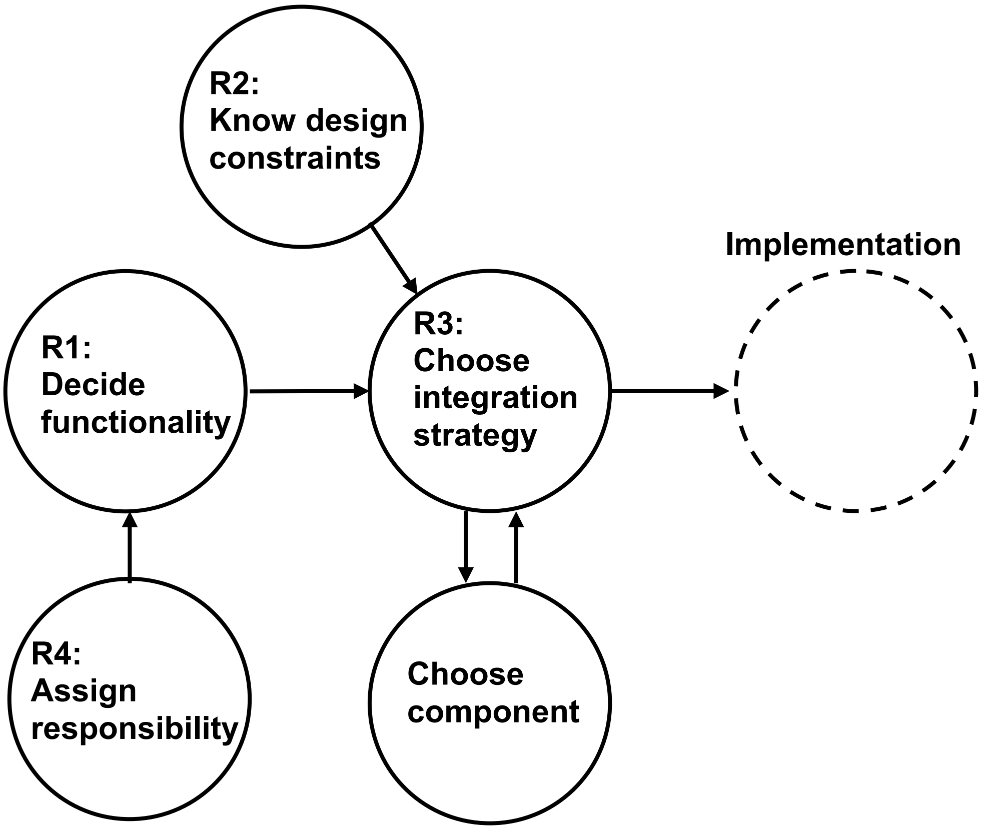

Figure 4. Sequence of recommendations

A recommendation, R1, on deciding functionality before starting integration can seem obvious, but we see that it happens in complex structures where the demand for functionality is distributed among departments. Failure to involve stakeholders and assign of responsibilities, R4, seems to cause erroneous or incomplete decisions on what functionality should be included. Thus, we deduct that R1 and R4 should be fulfilled before going into implementation.

Furthermore, the interviews revealed problems where the decisions on integration strategy were omitted. We deduct that integration solution can be done only when functionality is decided and thus we define that R1, and R4 should precede R3. If there are candidate components we argue that the best choice cannot be made without assessing the feasibility of integration for each one. R3 defines a set of decisions that should be addressed in order to make an optimal choice of component. If there is only one candidate, the decisions on strategy must still be made to avoid problems in implementation. The area of platform design constraints, R4, is needed in order to decide on integration strategy but seems unrelated to areas of functionality or responsibilities and thus we deduct that it should precede the integration strategy decisions.

These assumptions on the sequence of decisions were added when we expressed the recommendations. This is also how we measured fulfillment; we measured whether R1-R4 were fulfilled before going into the implementation phase.

Analysis after our validation study indicated that two more recommendations would have lowered project risk. R5 is related to checking that delivered components exhibit what is understood to be agreed. R5 is applicable in the implementation phase. R6 is a general advice that certain characteristics of projects could indicate increased projects risk. This advice has no position in the time graph, but could be applicable in a pre-study or component selection.

Relating this sequence to the gate model, we see that R1 and R4 is related to a pre-study phase or early concept phase. Function and responsibilities can be outlined in the pre-study phase but as concepts are chosen there may arise requirements that are more precise. R3 and R2, are applied in the concept phase when different concepts are compared. Going into detailed design in the implementation phase concepts should be chosen. However there is still decisions to make in the individual development areas and R3 and R2 are still applicable in electronics development, but should have been used mainly in concept phase.

2.1.3 Study C – Decision support model

Our third research question (Q3) is addressed by devising a combined model of ATAM, and AHP to support in decision making in integration projects.

Q3. How can decision making in integration be supported by structured methods?

Our study has resulted in a proposed model that can be used to guide design and integration. We have presented a new method for making decisions on integration strategy for in-vehicle automotive systems. The method is based on a combination of the Architecture Tradeoff Analysis Method, ATAM, and the Analytical Hierarchy Process, AHP. We have described the method in detail and exemplified its use with a theoretical but realistic example of an electronic controlled gearbox that is to be integrated into an in-vehicle electronic system. Analyzing the method and the example, we have shown that the method is usable and has benefits compared to either ATAM or AHP used individually. Like ATAM, this method provides a way for stakeholders to reason about system qualities, but it does not stop at identifying important design points. Compared to using ATAM alone, our combined method supports decision-making and should still have the benefits that have been reported with ATAM. One such important benefit is that stakeholders get to reason about qualities and their fulfilment. Thus, compared to using AHP alone, we will get both a structure for the criteria and likely also the benefit of stakeholder involvement and communication.

2.2 Applying the results

The potential gain for OEMs using our results lies in achieving more solid foundations for design decisions. Study A shows some of the import business drivers for designing automotive electronic systems. By using the checklists and recommendations of study B, an OEM could potentially find central decisions on integration strategy early that, if omitted, could cause delays. Thus, applying the result could avoid pitfalls and enable successful integration projects.

In addition, OEMs can elaborate on central decisions by using the proposed model of study C. Finding out the diverse requirements from the product life cycle is a prerequisite to making correct decisions. To actively agree on and estimate a candidate solution’s ability to fulfill these could be valuable in terms of product cost and quality. An OEM could potentially benefit from insight in the relative importance of requirements as well as insight in the reasons for choosing solutions.

Chapter 3.

Research method

Here, we describe the method of the studies we have performed. Study A and B are inductive studies where the data collection is more open ended and exploratory. We have performed case studies and collected data with which we have attempted to form theory.

Study C is more deductive in nature. We start by making theory, in this case a decision-making model for the problem, and then we move on to test it. The validity of deductive conclusions is either true or false whereas inductive conclusions can be supported to a degree. To show validity in study C, we describe our reasoning and assumptions.

For each of the three studies we describe the design of the study, the method used, and our reasoning on validity. We use four views on validity that each tests a different aspect of the quality of a study. These views have been described by Yin, Robson, and Wohlin [13][14][15]. The *construct validity* of a study is the certainty with which we can say that we actually measure what we want to measure. *Internal validity* is the certainty with which we can show causal relationships. Construct validity deals with the correct extraction of data, while internal validity deals with relations between data and the explanations of cause and effect. *External validity* or generalizability is the certainty with which we can say that a demonstrated result is applicable in other contexts. The *reliability* or conclusion validity of a study is the certainty that another researcher could perform the same study and get the same results. High reliability is achieved by defining the procedures of the study.

3.1 Study A – Business drivers and architectures

In our first study of driving requirements and contemporary design choices, the ultimate goal would be to achieve an optimal architecture design process where the quality requirements and their dependencies are fully understood and where the impact of choices in architecture, technology, and methods are also completely captured.

In order to take a step in this direction, we have defined the research question Q1.

Q1. What design drivers do exist in the business situation of OEMs and how do they affect electronic architectures in practical cases?

In order to answer this question we want to explore the business situation and the chosen electronic architectures of OEMs. There is also an explanatory part to this question as we would want to explain the relationship between drivers and architecture. We want to survey the use of architectures, methods and technologies in some automotive applications. The intention is to identify and analyze the requirements stemming from each business case and also examine issues of architecture, technology, and methods in each organization.

Hence, the intended objective of the study is to;

1. Investigate the business situation for organizations including factors of product volumes, valued quality attributes, the most important business processes. Investigate architecture, technology, and methods.
2. Analyse how the business situation of each organization are reflected in their respective architecture, choices of technology, and methods.
3. Analyse the mapping between requirement and solution, and provide a guide for assisting in selection of architecture, technology, and methods.

Method

Thus, an exploratory and explanatory study is wanted. The question of what drivers exist would be possible to answer by a survey. The question of causal relationship requires more to the study. We used the research strategy of a case study with survey and workshop elements. We collected data to explore the contemporary situation by asking practitioners. In addition, we used the method of workshops where the specialists could discuss causal relations. We then used the compiled data together with the suggested propositions of specialists to analyze and explain the relationship.

We used a series of semi-structured interviews with four experienced system designers. One from each manufacturer; Volvo Car Corporation, Volvo Trucks, Volvo Busses, and Volvo Construction Equipment. The data collected from the interviews was validated by the respondents.

We used questions on the topics; company context, functionality, cost, standards, and architecture. We held discussions on each topic and this resulted in data to lead our analysis. The interviews were executed in a number of workshops and the respondents were given assignments of finding data between sessions.

Construct Validity

In study A, our study is made by interviews with one system architect per company and the respondents reviewed the answers. There was also documentation to study. One threat to validity in our study is the use of only one architect per case. A misconception from one architect could cause a misreading of data. For instance, a reportedly important design requirement could be unimportant or the architecture was not in fact designed as stated. We addressed this by cross-checking with documentation as well as having open meetings where the architects openly scrutinized each others statements.

Internal Validity

In study A, we indicate which business drivers cause OEMs to make certain architectural choices. How can we know that these drivers in fact cause these choices and not an unknown and unmeasured parameter is in fact the cause? We argue that the listed factors in business situation are in fact part of the cause based on two things. First, system architects indicated some of the causal relationships. Secondly, our analysis supports the argument of the causal relationship.

External validity

Our conclusions from this study include description of contemporary business situation and architecture, as well as statements on causal relationships in the four cases. The topics for business situation and architecture could be applied in another context to aid in analysis. Applying the explanations cause and effect is difficult in another context. The design of architecture in other cases would probably follow the same lines of reasoning, but there may be other important business drivers that invalidate our explanations. However, the descriptive results of the study can be used as a basis for analyzing any case.

Reliability

In study A, we used open ended questions on topics that were chosen by us. The reliability of this approach is not great since the result could be influenced by the observer’s feel for what is enough information.

Therefore, there is a possibility that another researcher could get more or less data from doing the same study. However, another researcher would not likely, we argue, find different business drivers or architecture if using the same topics. Using documentation and specialist discussions, we have strengthened the confidence of getting the right data. Instead, a different researcher would perhaps find more or less information from the same study. We argue that this is not a large threat to reliability or validity. Observer bias is also a threat to reliability. My and our own opinions could color the analysis. In this first study, we did not make any claims on what is better in any way. Our analysis was instead focused on finding the factors that drive architectural design and therefore the threat to validity should be small.

3.2 Study B – Integration project success

In our second study of integration practices we would ideally want to have predictions on which practices and integration solutions that lead to successful projects. We have focused on the activities preceding integration and we define our second research question (Q2):

Q2. What key factors lead to success in integration projects?

In order to identify factors we need to know why projects fail. Yin [13] proposes typical research strategies for different types of research questions and characteristics of the studied phenomenon. A "what" question on a contemporary event without control over affecting parameters would indicate the use of a survey or documentation study. Solely using a survey would not, however, reveal all factors. The key factors are hidden in the explanations to why integration projects fail. Thus, there is an element of "why" in finding the key factors. We want to explore the problems, relate problems to affecting factors, and provide a remedy. We choose the strategy of a case study for exploring problems and identify possible relations to factors. We use a multiple case study to explore what is common problems.

In addition to illuminating the problems and influencing factors, we must define success to be able to answer question Q2. We define project success as completion on time and budget as they were estimated at the time of choosing the component. The strategy at this point would include what qualities and functions should be supported and changes of this strategy would likely show up as delays or added costs. The drawback of this definition is that we do not get a notion of the outcome in terms of how well the component meets its requirements; its quality. It would be theoretically possible to complete a project on time and budget but deliver a useless subsystem. This is not possible in practice, we argue, due to testing, and reliability growth programs. Problems in fulfilling the decided strategy would not be accepted by stakeholders such as the service organization and the project would be forced to solve problems. We perform the study under the assumption that time and cost is sufficient measures for project success.

Thus, the objective of the study is to show which factors are the most important. The idea is to provide, not only, a list of affecting factors, but a list of key factors that should not be omitted when executing a project of mechatronic integration. The objectives of this study are:

1. To identify key factors that affect project success.
2. To propose a checklist with recommendations.
3. To validate the recommendations.

Method

First, three cases were selected based on availability and timing. We performed interviews with senior technical staff with different roles in each project involved in the three cases of integration. Project manager, electronics engineer, application specialist. In each project we interviewed three persons. Each respondent were interviewed for approximately 1,5 hours of open ended questions and the topics were; 1 General, 2 Specification, 3 Integration solution, 4 Verification, 5 Result, 6 Future. We were two interviewers and we each documented the interview and then compiled the results to one interview document for each respondent. There were no audio or video recordings. The results were put in a table, compared, and analyzed. This step yielded a list of problems and our analysis yielded countermeasures or recommendations that supposedly would counteract problems.

Second, in order to validate or list of recommendations we chose two more projects. Again the choice was based on availability and project status. We devised a set of questions to find out fulfillment of our recommendation in four areas. In addition, we defined project success and included that in our questions. Further, we devised questions on project context in order to find other affecting parameters, outside the control of the project personnel. We carried out structured interviews and followed up with email and phone calls until all was answered.

Some non-public documentation was provided during the interviews. This information however is not used in the reasoning we provide in the analysis, only for verifying the statements of the respondents.

Construct validity

In study B, we claim that we have identified some of the key factors to consider when executing an integration project. Here we did use several people with different roles in each project to assure construct validity. We also studied documentation and again we let respondents review their statements.

Internal validity

We claim that we have shown some of the key factors affecting the success of the investigated integration projects. The question is if we can show that these are in fact the ones and if they actually cause the shown outcome. We argue that we have shown internal validity base on three supportive arguments.

1. The factors were extracted from specialists involved. There can be more parameters that these specialists were not aware of but we believe we have enough respondents to argue that we have found some of the key factors.

2. The correlation between the factors and the outcome is good enough that we argue that these parameters must be among the key ones. If other, to us unknown parameters, were the most important we believe it unlikely the correlation would be as clear as shown in Table 2. (The same figures are shown in paper B – Table 11)

3. We point out that many other thinkable factors are likely secondary. For instance, the experience/knowledge factor of an involved engineer or manager could also counteract problems in integration projects. We argue that this is not a separate factor. We have listed decision that lead to successful integration and a skilled engineer would perhaps focus on precisely these decisions, which would not falsify our claim. Thus, the skill level is not a separate factor.

However, we recognize that there may be more factors and that we are not able to show any statistical measures on the relative importance of all the factors.

External validity

In study B, there are threats to validity. The results and recommendations found are derived from five cases from a single company. Although the stipulated problem of integration is the same in another company independent of context, the differing context can yield different importance to the factors found in this study and also show more factors. One factor that may have impact and is likely to be different is the architectural choices in the platform. Perhaps a different architecture would put focus on other decisions.

In response to the threat of studying a single company, we argue that the five cases are in fact very different. Three have suppliers outside Sweden (none is involved twice), all five involves organizations outside Volvo, only cases #3 and #4 have project managers in the same company – the remaining three are run by different companies inside Volvo. This leads us to conclude that the cases are not homogeneous with respect to company culture, nationality, and project context; and we argue this in favor of external validity.

The problems in integration are general to automotive OEMs, and we have demonstrated that our recommendations are valid to tackle the stated problems. But the severity of each problem may well differ in a different context, and our recommendations, although we show them to be necessary, may not be enough to counteract problems in a general case. It is however likely that many of our recommendations are in fact valid within many automotive companies, although dedicated studies have to be made to verify this.

Reliability

For the validation part of the study, we used structured interviews with defined questions. In addition, we followed the interviews with a survey for the remaining data. We did follow up with phone calls and email until all the data was collected. We believe another researcher would be able to extract the same data following the same procedure.

Participant and observer bias are two possible factors that could cause threats to reliability. I work in the electronic development department and as such there could be matters of observer bias. Trying to defend myself I argue that there are no claims made that indicate that I am in favor of any certain aspect or solution.

3.3 Study C – Decision support model

Aiming for a general decision support model, we define our third research question (Q3):

Q3. How can decision making in integration be supported by structured methods?

In order to get a general result, we perform a deductive study where we reason about the problems in deciding on integration strategies. Here, we want to propose a model to structure requirements and decision making in design of automotive electronic systems. Further we want to evaluate different integration strategies to find the one that best support the desired qualities of the product in its life cycle. In order to evaluate success of different integration strategies we need some criteria on how to decide what is successful. The approach of this work is to use scenarios from the Architecture Tradeoff Analysis Method, ATAM [6] to describe system goals, and evaluate candidate designs with the Analytical Hierarchy Process, AHP [7], to evaluate different integration strategies in the context of an automotive electronic system.

Method

The method to find a model to support decisions is based on our reasoning. The first criterion for a model, we argue, is that it should be as precise as possible. There should be no estimates unless it is necessary. Using quality requirements is imprecise but there is no obvious alternative. Ideally, there could be a project where all requirements are stated in a proper requirements specification, but we are not sure this is possible even in utopia. If that happens in a project our model could handle real requirements, but we believe the use of scenarios can, but does not have to, get as precise as possible in practical cases. Thus, scenarios are employed to describe all the uses of a system and scenarios are what we demonstrate our model for. After eliciting all the uses of a system, we want to get a correct measure of the importance of each scenario. For a complex system, this seems to us like a task where no exact answers exist. Asking many stakeholders from the life cycle of the product to grade importance by their opinion is a workable solution. Finding out how well a design choice fulfills a certain scenario could be inexact even in very simplified situations. Asking architects is one possible solution. Negotiating a weighted multi criteria decision with set of importance rated scenarios is however possible without estimations and AHP is a method for doing just that.

Construct validity and reliability

In study C, which is a proposed model for decision-making, there are no direct observations that need to be tested for validity. Instead, we describe our reasoning around the problem and we argue our reasons for selecting ATAM and AHP as a basis for our model. We support our reasoning with an example to demonstrate the use of the model.

Internal validity

We describe a decision support model and we show how it can be used for a practical case. We claim that it is valid in the sense that it has benefits compared to either ATAM or AHP used alone. We provide reasoning in support of using this method. However, the use of the method is not tested by us in any real decision making. We do claim that it is usable and sound based on reasoning. If there are stakeholders to express their requirements and there are architects that can estimate importance, then our model works.

External validity

How general is this model for decision support? The model is not dependent on context. There is no obvious reason to us it would not be general, but we have not explicitly validated the model in different contexts.

Chapter 4.
Related Work

In this section we present other work that has been done in the area of integration of automotive electronic systems.

4.1 Study A – Business drivers and architectures

Investigating automotive electronic architectures with the intent of providing a design guideline is a line of work with many related areas. It includes topics from such areas as requirements and systems engineering as well as software engineering. A guideline in design must also consider issues of technology. Automotive electronics includes technologies in many areas such as field busses, software components, hardware components and development tools. Available technologies exist for diverse purposes like diagnostics, communication protocols, simulation tools, by-wire applications, and much more.

Architecture

The architecture of a system is the structure and the principles behind its design and evolution [12]. Choosing architecture directly affects system properties and quality attributes. However, the relation between design decisions and the outcome in terms of quality is not well understood. There is a need for analyzing system fulfillment of quality criteria at the design stage. In study A we investigate architectures and their driving quality attributes.

Product line architecture, PLA, is proposed to architecturally model a complete set of products in a product line [21], to lower cost and increase quality. The PLA approach is also intended to increase understanding of the relation between design decisions and quality turnout, but focusing on an architecture encompassing several products. Case studies have shown benefits in development time and quality outcome of a product line architecture approach [22]. The product line situation matches well with automotive industry's way of building different products from the same platform or asset base. The approach put focus on that the platform has its own development cycle, which is longer than the cycle of developing products. For integration of electronic components it is important to recognize the constraints imposed by the platform.

Engineering challenges

Farbman et al. describes the challenges in engineering computer-based systems in [9]. Engineering of computer-based systems is associated with problems in many areas from both a systems engineering and a software engineering perspective. There are difficulties in eliciting, defining, and negotiating requirements [10] that are independent from design, especially when considering non-functional requirements. Our exploratory study is aimed at providing knowledge on the driving requirements and the resulting architectures in the domain of automotive computer-based systems. We use the identified problem areas of this work to study and describe the situation that automotive electronic engineers face.

4.2 Study B – Integration project success

Integration practices

Related to integration practices, integration in the automotive domain is the effort on joining mechatronic subsystems from many suppliers. To achieve success in such projects, use of sound systems engineering principles is a key to success. To assess a company's systems engineering capabilities the SE-CMM method can be used [23], the practices described in the method can serve as guidance when developing processes and guidelines. SE-CMM practice area PA 05, which was partially integrated into CMMI treats our area, system integration.

The SE-CMM PA 05 recommendation stipulates good practices and activities to carry out integration. These involve generic advice that need interpretation and whose fulfillment is difficult to measure. Our results from study B differs in that they explicitly identify which decisions are the most important for the domain of automotive electronic integration. We focus on describing the details of practical cases and assess which practices are the key factors in achieving success.

Engineering management

In a study by Nellore and Balachandra [26], factors that contribute to success in automotive development projects at a major European car manufacturer were reported. The study covers the entire project phase, and as a consequence, is not very detailed regarding subsystem integration. However, Nellore and Balachandras shows that supplier involvement is one of five key factors to achieve success. The results show that suppliers require different level of specifications depending on their history of OEM cooperation. We conclude in our study that a number of decisions must be taken in order to achieve success in an integration project, which is a part of an integrated product development project. Our recommendations include both management- and engineering advice but differ in that they are detailed and do not need interpretation to be implemented.

Software engineering in automotive context

Software engineering researchers has also recognized the problem of integration. Pretchner, Salzmann, and Stauner are organizers of the ICSE Workshop on Software engineering for automotive systems, a workshop that in 2005 were focused on integration. The organizers describe the specific challenges in integration: "Automotive systems consist of a number of independently specified and developed subsystems that have to be integrated into the automotive system. … Because of safety and quality requirements on automotive software, this integration phase is of particular relevance for software development." [3]

Technologies and standards

We focus on integration of the electronic part, i.e., software and electronics hardware, of the mechatronic component, because this part is often problematic to integrate. Thus, this is an area where more research is needed. There are two major design approaches to integrate the electronic part of a mechatronic component into the electronic system; we will examine both alternatives here:

1. Hardware integration, which traditionally is the most common approach in the automotive domain. The integration here is realized through connecting a computer system, i.e., Electronic Control Unit (ECU), to a computer network in the vehicle. The integration interface is the computer network, and interaction with other parts of the system is through message exchange over the network.

2. Software integration, which has much focus in current research and standardization efforts in the domain. In this case the integration is performed through deploying a software component on an ECU which is not part of the mechatronic component. The intention is to deploy several software components on the same electronic hardware; one goal is to decrease the number of ECUs. The software integration solution offer to partition software and configure ECUs more freely. In this case the integration interface is the software environment on the receiving ECU.

To simplify the specification and integration work with hardware components, standards can be used. Common in vehicle industry today is to use CAN [24] busses as physical medium with standard protocols on top, e.g., SAE J1939 [29] specifying the syntax and the semantics of certain messages including message identifiers and value range, as well as their meaning. In terms of integration this gives the OEM and suppliers a common agreement on bus interface. Another common method in combination with CAN is to use tools to package individual signals into proprietary messages on the bus and derive priorities so that temporal constraints are met, e.g., the Volcano tool [6]. In this case the OEM-supplier agreement is not standardized bus messages, but standardized communication software on ECUs which let the OEM configure bus traffic.

Standards have also been used to simplify integration of software components. AUTOSAR [31] is a European automotive industry initiative that aims at developing a standardized software architecture for software components including basic software functions such as communications and diagnostics.

In the automotive domain OSEK/VDX [28] aims to define an open architecture that standardize software interface to communication, network management, and operating system. There are currently much research efforts in the area of software component technologies for embedded systems; several of these results could be interesting to support integration of software components in our context. However, availability of suppliers supporting the technologies is a requirement by OEMs before adopting it as integration platform. A major ongoing project is the DECOS project [32]. DECOS is focusing on safety critical systems, e.g., automotive and aerospace applications. The core of DECOS is a time-triggered architecture, providing both spatial and temporal partitioning, preventing interference among components sharing access to devices, as well as timing interference between components. Integration of software components is recognized as a key challenge and the first of several DECOS objectives is the "development of a distributed execution platform that allows tight integration of software-modules from different sources (vendors) and with different criticality levels".

Standardized software and AUTOSAR is related to integration in that it explicitly addresses the problem of integration of software. The focus is on the technical solutions to enable integration and the architecture includes interfaces, communication and diagnostic principles, and tools. Our study is focused on finding key factors for executing integration projects. We recognize that a unified architecture could be one such factor. The problem of integration will not however be solved by the introduction of e.g. AUTOSAR. The results, conclusions and recommendations of our study are valid for integration of both hardware and software components.

4.3 Study C – Decision support model

Theory on architecture

Integration involves design decisions and results from the field of architecture analysis and design can be applicable. Integration is the design of the interaction between a component and its environment and choices in integration also affect system qualities. Kazman proposes Scenario based analysis of software architectures [18] to handle imprecise requirements and approximate system usage via scenarios. Several scenario-based evaluation methods [17] are proposed where usage scenarios are developed for theoretically trying out different candidate architectures described by some high-level design description. Architecture description languages, ADLs [16], are intended to provide the high-level description of the system and give analyzability to system properties.

In order to analyze an architecture and find risks and decision points which affect outcome of quality attributes, the Architecture Tradeoff Analysis Method, ATAM, has been proposed [6]. The important decision points in system design is reflected in designing integration solution.

We use the ATAM way of defining qualities, and then we use this to aid in choosing integration strategy. We base our model around the theory on architectural assessment. We incorporate views of the system to model the different stakeholder needs and we choose the use of scenarios to describe system uses. Instead of analyzing an existing system by scenarios, we model desired properties of the system to be able to compare alternatives.

AHP and CPC

The Analytic Hierarchy Process, AHP [7], is a multi-criteria decision making approach in which factors are arranged in a hierarchic structure. We apply this method on our weighted usage scenarios evaluate design alternatives that we define. In AHP the weighting of the factors is performed by comparing each factor to each other factor yielding a matrix of comparisons. Chain wise paired comparison, CPC[8], is a simplified way of comparing the factors where each factor is only compared with its next-in-line factor. Thus, instead of getting a matrix we get a chain of comparisons. We propose the use of CPC to reduce the number of comparisons that have to be done by specialists.

4.4 Summary

In summary, we see related work to design and integration of automotive electronic systems in several areas. The field of systems engineering have results in the field of handling complex systems with requirements engineering and other methods to handle the execution of complex projects. The field of software architecture design have results in areas of handling non quality attributes and analyzing architectures with respect to these. There are reference models for development processes that include guidance and best practices of integration. There are also technologies and architectures that explicitly address the problem of enabling feasible integration for systems such as vehicles. The closest thing to a guideline for structured engineering of integration is the process guidelines provided by reference models such as SE-CMM [23]. However, our results differ in that they provide concrete checklists whose fulfillment is observable. The scope of our results is for integration of electronic components in the automotive industry. In addition, we identify which factors are the most important ones and validate this identification with the intent of providing guidance as opposed to the reference model where a number of generic recommendations are to be followed.

Chapter 5.
Discussion and conclusion

In this chapter, we provide a short elaboration on the topic of automotive electronic integration followed by the conclusion of the thesis.

5.1 Discussion

The trend towards more system wide optimizations and more electronic control functions is strong. Consequently, the need for life-cycle related functionality increase with operational functionality. Complexity of the system will increase as a result. Integration in this context will become increasingly difficult unless there is some way of handling complexity.

Advanced behavior control in vehicles

We see an increase in electronic functions now and in the years to come. There is both an increase in the number of subsystems that include electronic control, and in the number of system wide functions to affect vehicle properties. Previously non-electronic domains such as brakes and steering see the introduction of electronic functions. Electric hybrid vehicles involve electrical motors for propulsion and auxiliaries, and this yields more electronic control functions. In addition, hybrids specifically require coordination functions in achieving optimized fuel efficiency. Active safety applications introduce electronically controlled devices such as radar and cameras as well as system functions such as brake initiation.

Hybrid vehicles are a good example of how the interactions can get more complex. For an electric hybrid vehicle, there are more mechatronic components too coordinate and the system property of energy consumption is explicitly addressed. The energy optimization of a vehicle needs to involve many components including engine, electric motors, energy storage, brakes, pedals, gearbox etc. A low energy consumption is in direct conflict with e.g. performance and handling. In order to accomplish this, the resulting system involves more interaction. Each component must be made to allow the vehicle system to change its behavior in a way that enables a global energy scheme.

At the same time as this increase in functions, there is a trend towards fewer and more powerful ECUs. Moore's law shows that the cost of electronic components such as CPU and memory will decrease but the cost of cables, and ECU housings will not. Consequently, ECUs with more powerful processing could host the increased number of functions without increasing the product cost. Hosing more functions in a single ECU means that more effort will go to areas of software infrastructure. The automotive electronic development is much concerned with safety and reliability of functions and assuring integrity of functions.

Integration in this context

An increase of electronic functions causes an increase of electronic system complexity. This, in turn, causes a need for more aspects of components to be considered in component interaction. This will accentuate the problems that we report in study B and integration will become more complex. Handling the complexity is a challenge and a key to the solution. The provided recommendation is a step towards performing integration projects, but is not a definitive solution to the problem.

The problem of integration is perhaps a wicked problem [33] if we attempt address all problems at once. Applying the concept of views [34] is one possible tool as a means to separate concerns. The concept of an information model [35] is appropriate to allow modeling of all different views of the system. A model could theoretically assist or even generate design, but we believe information model yields another immediate result on integration problems. That is to achieve consistency in information during the development. If all developers see information in some repository, there would be less room for erroneous interpretation and old assumptions to yield wrong design decisions.

5.2 Conclusion

In this thesis, we have presented our results on integration of electronic components in vehicles. Our aim has been to provide knowledge on how to integrate automotive electronic systems successfully in a setting where vehicles are developed based on existing platforms. Our focus has been the early phases of automotive electronic system development and in particular on the decisions taken in integration of electronic subsystems. Figure 5. below is a copy of Figure 3 in the Results section and shows where the results are applicable in a generic development process. The figure shows that integration takes place late in the detailed development phase and our results demonstrate that successful integration is achieved by decision making in the phases prior to integration. In summary, the results are aimed at supporting decisions in early phases of development including concept and strategy choices.

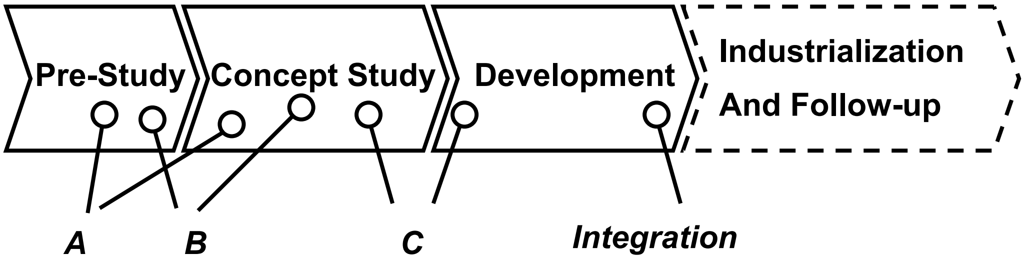

Figure 5. Applicability of results

We present three results that support decision making in integration of automotive electronic systems. The results correspond to the three studies; study A, study B, and study C.

The results in study A illustrate some of the design drivers that exist in the business situation of automotive OEMs. For the four studies cases we provide an analysis of how these design drivers affect design and choice of architecture. An important lesson from study A is that one should be very careful to uncritically apply technical solutions from one industry in another, even when they are as closely related as the four vehicles describes in study A. Understanding the requirements from the business situation is the key to choosing architectural solutions.

Study B presents validated recommendations for performing integration in automotive development projects, each including a set of checkpoints that defines recommendation fulfillment. We identify that the root causes of problems in integration are largely related to decisions omitted in electronic strategy. Our recommendations are defined by checklists for critical decisions in the areas; functionality, platform, integration, and stakeholder involvement.

Study C demonstrates the use of a method for supporting decision making when considering integration alternatives. Analyzing the method and the example, we have shown that the method is usable and has benefits compared to either ATAM or AHP used individually. We argue in benefit of the method; Firstly, it is scalable in effort to compensate for more or less crucial decisions. Secondly, we show that it provides feedback on the quality of the estimates. Thirdly, the method does provide some documentation as to why a decision has been made and this possibly helps in understanding and communicating system design among stakeholders.

Summary

In summary, the problem of integration in automotive electronic systems is shown to affect the ability of OEMs to complete products on time and cost. We have provided aid for engineering in the phases prior to integration with the intent of lowering the risk of failure; thus increasing the chance of success.

Chapter 6.
References

[1] Leen G, Heffernan D, Expanding automotive electronic systems. IEEE Computer 35(1):88 – 93, 2002.

[2] Jakob Axelsson. Methods and Tools for Systems Engineering of Automotive Electronic Architectures. In Proc. Design Automation and Test in Europe Conference (CD-ROM version), Munich, March 13-16, 2001.

[3] Alexander Pretschner, ETH Zürich, Switzerland, Christian Salzmann, BMW Car IT, München, Germany, Thomas Stauner, BMW CarIT, München, Germany, 2nd International ICSE Workshop on Software Engineering for Automotive Systems, St. Louis, May 21, 2005,

[4] Auer G (2004) Automotive News Europe, Mercedes will make cars less complicated, technology leader cuts 600 electronic functions, seeks more. May 31 2006, http://www.accessmylibrary.com/coms2/summary 0286-21549689 ITM (Last Accessed: 2006-10-17)

[5] Axelsson J (2006) Cost models with explicit uncertainties for electronic architecture trade-off and risk analysis. In: Proceedings of The International Council on Systems Engineering (INCOSE) 2006.

[6] R. Kazman, M. Klein, and P. Clements. ATAM: Method for architecutre evaluation. Technical report, Carnegie Mellon Software Engineering Institute, CMU/SEI-2000-TR-004.

[7] T. Saaty. The Analytic Hierarchy Process. New York: McGraw-Hill, 1980.

[8] J. W. Ra. Chainwise paired comparison. Decision Sciences, 30(2):581–599, 1999.

[9] Engineering computer-based systems: meeting the challenge, Farbman White, S., Melhart, B.E., Lawson, H.W., Computer, Volume: 34 Issue: 11, Nov 2001, Page(s): 39 -43

[10] Boehm, B., Egyed, A., Software requirements negotiation: some lessons learned, Software Engineering, 1998. Proceedings of the 1998 (20th) International Conference on, 19-25 Apr 1998, Page(s): 503 - 506

[11] Stage gate model, www.stage-gate.com, (accessed: 20071101)

[12] IEEE Recommended Practice for Architectural Description of Software-Intensive Systems. IEEE Std 1471, 2000.

[13] R. K. Yin, Case Study Research: Design and Methods, 3rd ed: Sage Publications, 2003.

[14] C. Robson, Real World Research, 2nd ed: Blackwell Publishers, 2002.

[15] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, Experimentation in software engineering; an introduction. Norwell, Massachusetts, U.S.A.: Kluwer Academic Publishers, 1999.

[16] Paul C. Clements, A Survey of Architecture Description Languages, Eighth International Workshop on Software Specification and Design, Germany, March 1996

[17] Muhammad Ali Babar, Ian Gorton, Comparison of Scenario-Based Software Architecture Evaluation Methods, Proceedings of the 11th Asia-Pacific Software Engineering Conference (APSEC '04), 2004

[18] Kazman,R., Abowd, G., Clements, P., Scenario-based Analysis of Software Architecture, IEEE Software archive, Volume 13 , Issue 6 (November 1996) table of contents, Pages: 47 – 55, 1996, ISSN:0740-7459

[19] D. Garlan, What is Style? Proceedings of the First International Workshop on Architectures for Software Systems, pages 96-100, April 1995.

[20] Bass, Clements, Kazman, Software architecture in practice, Addison Wesley, 1997.

[21] Clements, P., Northrop, L., Software Product Lines, Addison-Wesley, August 2001. ISBN 0-201-70332-7.

[22] Brownsword, L., Clements, P., A Case Study in Successful Product Line Development, Technical Report CMU/SEI-96-TR-016, October 1996.

[23] Electronic Industries Alliance (EIA) (2002) Eia/is-731.1, systems engineering capability model, electronic industries alliance (interim standard)

[24] International Standards Organisation (ISO) (1993) ISO 11898 road vehicles - interchange of digital information – controller area network (CAN) for high-speed communication

[25] Kazman R, Klein M, Barbacci M, Longstaff T, Lipson H, Carriere J (1998) The architecture tradeoff analysis method. In: Proceedings of the Fourth IEEE International Conference Engineering of Complex Computer Systems (ICECCS '98), IEEE

[26] Nellore R, Balachandra R (2001) Factors influencing success in integrated product development (ipd) projects. IEEE Transactions on Engineering Management 48

[27] Nellore R, Söderquist K, Eriksson KÅ (1999) A specification model for product development. European Management Journal 17:50-63

[28] OSEK/VDX (2006) Osek/vdx operating system. URL: http://www.osekvdx.org/mirror/os222.pdf

[29] The Society of Automotive Engineers (SAE) (2004), Truck and bus control and communications subcommittee, SAE j1939 standards collection

[30] Casparsson L, Rajnak A, Tindell K, Malmberg P (1998) Volcano - a revolution in on-board communication. Tech. rep., Volvo Technology Report

[31] AUTOSAR (2006) Homepage of automotive open system architecture (autosar). URL: https://www.decos.at/ (Last Accessed: 2006-01-17)

[32] DECOS (2006) Dependable embedded components and systems. URL: https://www.decos.at/ (Last Accessed: 2006-01-17)

[33] Rittel, H., and Webber, M.; Dilemmas in a General Theory of Planning, pp 155-169, Policy Sciences, Vol. 4, Elsevier Scientific Publishing Company, Inc., Amsterdam, 1973.

[34] Kruchten, P., Architectural Blueprints—The “4+1” View Model of Software Architecture, IEEE Software 12 (6), November 1995, pp. 42-50

[35] Chen, D., Törngren, M., Shi, J., Gerard, S., Lönn, H.,Servat, D., Strömberg, M., Årzen, K., Model Integration in the development of Embedded Systems – a characterization of current research efforts, IEEE International Sysmposium on Computer-Aided Control System Design 2006, Munich, Germany. October 2006

Part 2

Paper A

Business Situation Reflected in Automotive Electronic Architectures: Analysis of Four Commercial Cases

Joakim Fröberg, Kristian Sandström, Christer Norström

Presented at SEAS,
St Louis, USA, 2005

ABSTRACT

Automotive vehicle electronic systems are developed facing a complex and large set of inter-related requirements from numerous stakeholders, many of which are internal to the Original Equipment Manufacturer, OEM. The electronic architecture, of the product, or its structure and design principles, form an equally complex construct; including technology and methods, which ultimately should be chosen to optimally support the organization's own business situation.

In this paper, we have analyzed the relationship of four automotive electronic architectures to their respective business requirements and business context. The study shows four functionally rather similar products with computer controlled power train, body functions, and instrument. In the light of the business situation, we explain the solutions and why design principles are pursued. The analysis shows that despite a common base of similar vehicle functionality the resulting electronic architectures used by the four organizations are quite different. The reason for this becomes apparent when looking at different business context and business requirements and their affect on the architecture. Differences in business situation drive the use of different methods for integration, different standards, different number of configurations, and different focus in the development effort. Some key parameters in business situation affecting architectural design decisions are shown to be product volume, size of market, and business requirements on openness and customer adaptation.

An important lesson from this is that one should be very careful to uncritically apply technical solutions from one industry in another, even when they are as closely related as the applications described in this work. Understanding the requirements from the business case is the key to choosing architectural solutions.

Keywords

Automotive, Electronic Architecture, Case-Study, Requirements

1. INTRODUCTION

Designing a complex computer system such as an in-vehicle electronic system is a process of choosing solutions that best meets the huge set of, often conflicting, requirements. Modern in-vehicle electronic systems must provide functions and exhibit properties to support several of the OEMs business processes. In fact, the main part of the requirements does originate from the OEM business processes such as production, aftermarket support, variant handling, verification, and commonality efforts. The desired functions can be very different in nature, and the desired properties can be conflicting. Functional solutions span from web- to control applications and the desired properties call for radically different architectures and technologies.

Thus, the automotive industry seeks an improved way of synthesizing all the requirements into an electronic architecture that meets the diverse requirements from the business case as closely as possible.

In this paper, we present key findings from four case studies with the intention of describing the situation for commercial vehicle electronics developers; both the diverse requirements and the solutions in terms of architecture; as well as analyzing the relation between requirements and solutions. The inspected electronic architectures are all Volvo brand vehicles; Volvo Construction Equipment (VCE), Volvo Trucks (VTC), Volvo Busses (VBC), and Volvo cars (VCC). The three first are companies within the Volvo Group, and Volvo Cars is a subsidiary of Ford Motor Company.

Out of the complete result of the study, we have listed the key figures into tables and attempted analyzing the relations. Further, with the driving requirements in mind, we have commented on how several of today’s trends address the studied OEM challenges.

The first contribution of this paper is the analysis of how key parameters in business situation affect OEM choices in architecture. The second contribution is the analysis of the relation between OEM business requirements and some of today’s trends in automotive industry.

Section 2 contains the key figures of the study (2.1), the analysis of relations between architectural solutions and key figures (2.2), analysis of other findings (2.3), and summary of analysis (2.4). Section 3 presents comments to some of today’s trends in automotive electronics development with respect to the requirements outlined in this study. Section 4 concludes the paper.

2. THE FOUR CASES ANALYSED

The four cases were investigated with respect to background, functionality, cost, standards, integration, and architecture. Informants from the four organizations were interviewed in a series interviews and common workshops where all informants participated. The complete data from the study is presented in [1]. Here, we outline the characteristic findings from each case.

2.1 Key Figures from Study

In order to compare the cases and analyze the result we have extracted a number of key parameters in business context, business requirements, and resulting architecture from the case studies and listed them in the following tables. Using this data, we present analysis of the correlation between key parameters in business context, business requirements and electronic architecture solutions.

Table 1. Business context for each organization

Organization **Business context**	**VCE** **Constr. machines**	**VTC** **Trucks**	**VBC** **Buses**	**VCC** **Cars**
Production volume	~15000	~80000	~9000	~400000
Products	~35	~8	7	~8
Vehicle platforms	4	3	2	3
Organization size electronics	~45	~140	~30	~400
Market share	~5%	~15%	(~15%)	~1%

VTC product volume includes only the Volvo brand trucks. ‘Products’ is the number of models that have an own model name. ‘Vehicle platforms’ is the number of physical platforms used to achieve all the products. The ‘organization size’ includes the number of people who are working with development of electronic systems. The ‘Market share’ measure is an estimate of the percentage of the market that the OEM is in; the whole markets of construction equipment, trucks, busses, and cars respectively.

The Volvo bus figure of 15% is related to only the European market, which is VBCs strongest market, and the percentage of the world market should thus be considerably lower.

Table 2. Business requirements for each organization organization

Organization Business Requirements	VCE	VTC	VBC	VCC
Product variants	Few	Very many	More than very many	Many
Commonality	High	High	High	High
Hardware optimization	Low	Medium	Medium/ Low	High
Openness	Some	High	High	None
Customer adaptation	None	Much	Very much	None
Safety critical	Yes	Yes	Yes	Yes
Advanced control	Yes	Yes	Yes	Yes
Infotainment	None	Some	Some	Much
Telematics	Little	Much	Much	Some

Table 2 constitutes the key business requirements for the different organizations as elicited from the case studies. ‘Product variants’ indicate the diversity of vehicles requested by customers. ‘Commonality’ is the focus of the own organization to commonalize components between products. The requirements for ‘Hardware optimization’ are an estimate of the level of optimization that the organization desires for target products. ‘Openness’ reflects the requirements on ability to be open and integrate vendor components such as an engine Electronic Control Unit (ECU). ‘Customer adaptation’ refers to the whishes of customers to add or change functionality (often by adding ECUs) to the existing system. ‘Safety critical’, ‘Advanced control’, ‘Infotainment’, and ‘Telematics’ measures represent relative estimates on the requirements for the respective functionality.

Table 3. Architecture solutions for each organization

Electronic Architecture	**VCE**	**VTC**	**VBC**	**VCC**
Physical configurations per product	Few	Very many	More than very many	Many
Network information	Moderate	Very large	Very large	Huge
Standards – network application level	J1587 – J1939	J1587 – J1939	J1587 – J1939 Proprietary (Volcano)	Proprietary (Volcano)
Network technologies	2	2	2	~4
Internally developed nodes	All – 2-5	Few - 4-5	Few - 2-3	Very few – 1-3 (partly)
Ext node suppliers	0	~6-8	~6-8	>10

Table 3 presents the architecture solutions used by the different organizations. The ‘Physical configurations per product’ is the number of possible variants in ECU, sensor, and actuator configuration. The ‘Network information’ is the amount of information on the vehicle network. ‘Standards – network application level’ denotes the standard used for specifying syntax and semantics of network messages on the application level. The number of ‘internally developed nodes’ refers to nodes whose functionality is implemented internally and not necessarily the hardware. In the VBC case, the number of internal nodes is what is developed for the chassis, and the number gets higher if VBC also develops the bus body.

2.2 Analysis of Architectural Solutions

Here we analyze the parameters presented in the architecture table (Table 3) in relation to the business context and business requirements.

Physical configurations per product - A high number of variants in physical products is not something that an OEM desires. The aim is always to keep the configurations as few as possible to ease operations and thereby lower cost.

The VTC high number of physical configurations is likely in correspondence to requirements on openness and customer adaptation. VTC customers require very high openness of system with a configurable drive train that can include non-Volvo engines and gearboxes with non-Volvo electronics. Further VTC delivers to body builders as indicated by the high measure on customer adaptation.

VBC has a similar situation as VTC, but with even higher demands from customers that are body builders and add chassis and superstructures to the vehicle. The superstructures include much electronics and this drives a need for numerous interfaces to the system delivered by VBC. Thus, VBC shows an even higher measure than VTC in the number of physical configurations.

VCC that does not have high requirements on openness and customer adaptation, show a smaller number of configurations, but still VCC has many configurations. This high measure has more to do with the requirement for hardware optimization as many configurations can provide just that. The high VCC product volume is the underlying factor producing the high requirements for hardware optimization.

VCE who has neither the volume to drive requirements for high hardware optimization or the direct requests from customers to provide an open system, shows no variants in ECU configuration and only few sensor/actuator configurations. Instead, VCE has a relatively high number of vehicle platforms and products, which suffices to provide a sufficient number of configurations to meet customer requirements.

Network information - VCC show the highest amount of network information. VCC also shows requirements for much infotainment functions and some telematic functions, which partly explains the high amount of network information. Even so, the stringent requirement on hardware optimization is likely to affect this measure. Physical components can, to some extent, be replaced or reduced by the use of computer functionality and, thus, reducing product cost and weight of the car. The product volume and the requirements for hardware optimization amplify the arguments for introducing these functions as they become available and, thus, yield an increase in network information.

The amount of functionality can be expected to be in relation to the amount of network information. Thus, supposedly the requirements for functionality, including safety critical control, advanced control, infotainment, and telematics, drive the amount of network information. The requirements for hardware optimization are, in this sense, requirements for functionality that removes or reduces physical components.

This reasoning corresponds well to the situation of VCE, and VTC, which have moderate and vary large amounts of network information respectively. VBC, however, shows a low volume and correspondingly low requirement on hardware optimization, at the same time as having equal requirements on functionality as VTC otherwise. The explanation for this seems to be the tight relation between to VTC with many systems reused.

Standards – Network application level - VTC customers require freedom of choice in use of non-Volvo engines and gearboxes that come with ECUs and network interfaces. SAE J1587 was the used standard for diagnostics in the US market and is therefore required to be supported [4]. Because of the situation with vendor ECUs and body builders, the distributed applications cannot be governed by a VTC specific method. SAE J1939 is a standard that addresses problems in integrating ECUs from different vendors in that it defines syntax and semantics of signals.

VCC is a passenger car company and that segment of the vehicle industry does not have standards that cover OEMs and suppliers, because car customers do not require the ability to integrate a certain vendor engine. Instead, VCC is free to choose tools and methods to accommodate a network application level interface as seem fit. VCC uses the Volcano concept for two reasons: (1) Volcano supports integrating vendor ECUs while allowing VCC to manage the network traffic. (2) Volcano also facilitates automated optimization of network usage by packing signals into frames to save bandwidth with guaranteed timing. (1) is desired because of the high number of external node suppliers. Having a communication component that provides communication services as specified by VCC, provide management of a network with many different ECUs. (2) is desired because it provides VCC with good control of bandwidth and timing which, in turn, provides benefits. Firstly, high network efficiency addresses goals in hardware optimization which is high in the VCC case. Predictable timing is beneficial for developing and assuring safe and reliable functionality.

Network technologies - The large amount of information together with the requirements for optimization in the VCC case, imply that using several tailored networks for specific needs can be worth the added development effort. The use of LIN networks [2] provide a cost effective network for handling locally interconnected lights and switches, and a high bandwidth MOST [3] network serves the needs of infotainment applications.

VTC with a relatively high product volume has not chosen to introduce low cost or infotainment networks. Evidently, the benefits have not been deemed large enough for these specialized networks compared to the development cost and increased complexity of the system. Also for VBC and VCE, the increase in development cost for designing tailored networks is deemed unprofitable and this is reflected in the small number of network technologies.

Although additional network technologies mean added complexity, LIN for example can lower complexity due to its ability to achieve variants without the need for ECU I/O variants or software variants. Also, as is the case with VTC, VCE, and VBC there are commonality goals within the organization that strongly affect the choices in network technologies.

Internally developed nodes - The number of internally developed nodes differs in the four cases. VCC shows very few internally developed nodes, while VCE develops all nodes internally. The reason for this difference is mainly the differently sized markets. The market share of each organization together with the product volume shows an indication of the total size of the market. VCC stands out as operating in a very large market (~1% with ~400000 units). The size of market creates a situation where suppliers can accommodate many OEMs and get a huge market. This, in turn, yields prices that are, in many cases, considerably cheaper compared to OEM internal components.

For VCC, this means that developing components is sometimes not an option as it would be a considerably more expensive alternative. Also, the fact that VCC shows business requirements on infotainment e.g. video, games and communication makes VCC prone to purchase these systems as they are often produced for the large mass market of consumer electronics.

VBC, on the other end, shows the smallest market (~24% with ~9000 units), and the potential for suppliers to gain large markets within the bus segment would therefore be limited, if the bus segment of the market was isolated. Busses however, have numerous components that are similar to the truck market. This and the fact that VBC and VTC are so tightly related makes the VBC measure of few internally developed nodes, difficult to interpret.

VTC, shows a relatively large market (~15% with ~80000 units), but still orders of magnitude smaller than VCC, develops a minor part of the ECUs internally. Apparently, the price benefits of purchasing supplier ECUs are not as great as for VCC, due to the smaller market. However, in terms of electronic systems, the truck market is also closely related to other markets e.g. busses, and this makes the potential market bigger for suppliers.

VCE, who shows the second smallest market (~5% with ~15000 units), develops all ECUs internally. The size of the market for similar components is too small for suppliers to produce at a considerably lower price. Even though the VCE market is not magnitudes smaller than that of VTC, the similarity between products in this market is questionable. The needed electronic functionality of a wheel loader is not necessarily related to that of an excavator for instance, and thus, a supplier does not easily target all products in this market. This fact is likely to affect VCE in the direction of choosing internal development.

The fact that cars have the by far largest market yields a situation where OEMs of other vehicles very well might consider using car components as their price is attractive, even though, they may not be perfectly suited to the intended application.

The key to explaining the differences in internal development between the four organizations is, thus, the size of the market of similar components. A supplier that can target many OEMs with similar needs in electronic functionality can achieve a market far larger than the OEM alone.

This general reasoning does not apply to all types of electronic functionality and all ECUs. Some components might address the whole vehicle market, while other may serve only a small fraction of the market. There are even areas where suppliers of electronic functionality can be target markets outside the vehicle segment such as machinery, consumer electronics etc. but the size of market does have the influence of creating cheap components and thereby making OEMs purchase rather than develop components.

2.3 Analysis of Other Key Mechanisms

The analysis of the resulting architectures against a background of business context and business requirements has shown a number of central mechanisms that are crucial to the reasoning of the OEMs. These key notions deserve some explanation.

Annual production volume - The case study has shown that the product volumes are different in the four organizations, and thereby also the focus on fixed cost and hardware optimization. The willingness to reduce variable cost at the expense of fixed cost increases with the product volume. One way of reducing variable cost is to optimize vehicle hardware content to include a minimum of resources. This way, development effort is spent to reduce the cost of each product. This is also reflected in table 1 by the organization size; VCC having the highest number of engineers in electronic development. Software components are not subject to the optimization profit in that they represent almost only a fixed cost. VCC that produces vehicles in the range of 400000, can benefit to a larger extent by reducing variable cost, and therefore an increased cost for design of optimal hardware is more profitable than for VBC that has volumes in the range of 9000.

The focus on commonality - The desire for ‘Commonality’ is the desire to commonalize and coordinate use of components in many product lines. All four organizations emphasize the desire for commonality, which shows that commonality is not solely related to the product volume. All OEMs desire commonality because of the benefits in purchasing large volumes of components, but also has to balance these goals with benefits of optimization to reduce cost. The reason for the shared emphasis, although volumes are different, is related to the fact that production and service is costly with worldwide distribution as well as factories and service shops keeping physical components in store. Hence, the number of physical components must be kept low. Software on the other hand, should not present a high cost for distribution and storage. Instead, the use of numerous variants of software puts strain on working process and configuration management, but not on the cost of operations.

Commonality also indirectly affects the use of technology, process and tools which should affect development cost, knowledge transfer and supposedly product quality.

Methods for integration - VCC, VTC, and VBC uses the communication busses as interfaces in the process of integrating subsystems while VCE is not yet integrating vendor ECUs at all. The method to perform integration differs between the organizations. The method of specifying bandwidth and signals with Volcano together with statecharts, and power consumption, is suitable if the vendors can agree to follow OEM specifications. VCC specifies in this way to vendors, while VTC and VBC both have requirements on high openness in that specific components should be possible to integrate. Some crucial components such as a vendor engine can be manufactured by a large company that does not easily conform to VTC or VBC specific requests. Instead the interfaces are defined in standards. This is, in short, how the different organizations use different methods for integration.

2.4 Summary of Analysis

The bottom line of the provided analysis is that, even though the four electronic architectures are used for vehicles with many similarities in functionality, the resulting architectures show differences in key architectural solutions. These differences stem from the fact that business context and business requirements differ in the four organizations.

Analyzing the relation between key parameters in business context, business requirements, and resulting architectural solutions has shown that the four organizations are choosing different architectural solutions. The key parameters that affect these choices are product volume, market size, and requirements for openness and customer adaptation.

These results are valid for the four organizations and for organizations with similar business situations. An automotive organization with some business parameters way outside the scope of these cases might not be included by the explanations provided. On the other hand, none of the lines of reasoning are specific to these four cases, except the commonality relation that exists within the Volvo group. Also, the line of reasoning is presented so that deviations from the assumptions in this work should be identifiable. The reasoning on basic parameters such as product volume, and market size should be applicable in a more general setting than just the automotive industry since these business settings has no dependency to automotive products.

3. TRENDS IN PERSPECTIVE OF STUDY

Against the background of this study, we use this chapter to reflect on some contemporary trends in automotive electronics development today. This constitutes discussion topics and speculation on why certain solutions are in focus today and presenting solutions in the light of some of the key challenges.

3.1 Summary of Requirements

In order to summarize some of the challenges faced by OEMs with respect to computer systems, we note that the following areas are recognized by the four organizations in this study.

Integration - The OEM situation puts integration in focus. The OEM must purchase components from suppliers in order to keep costs down, while at the same time leveraging reliability and safety. Methods and tools for specifying and verifying compositions are strongly in focus. Today, integration is largely done using a communication bus as an interface between vendor ECUs.

Cost, Safety, and Functionality - Drives the exchange of physical components to computer systems. Cost can be severely reduced by removing or reducing mechanical components e.g. the removal of steering column or reducing dimensions of a shaft. As more and more control is done by computers, optimizing or coordinating functions gets feasible. For instance; fuel consumption can be reduced by considering many temperature and load sensors, or brake coordination. Also, safety functions are made feasible by computers and software. While allowing functions such as ESP and active collision avoidance, computer controlled systems can also impose a challenge with respect to safety. Assuring computer system function is recognized as more difficult than assuring the replaced mechanical system.

Aftermarket - As the computer system become more complex, the handling of configurations gets more difficult. Functions to accommodate e.g. emission reduction or reduced wear, may require unique software or parameters for each individual vehicle. Moreover, keeping track of compatibility among the subsystems is a challenge since products live for a relatively long time with many versions released. Distribution and storage of software is not burdened with the high costs of physical components, but complex processes introduce cost and some risk as failures affect customer relations just as a failed physical component. Finally, the manufacturing of processors and memory chips may be discontinued during the vehicle life-time. This can force redesigns of hardware, causing costly re-verifications or costly stock piling of components.

Variants, Brand and Commonality - Requirements on providing computer systems in many variants yielding different look and feel of the product are recognized as important means to satisfy different customers. Achieving this by using variants of the same design is desired due to goals in commonality.

3.2 Addressing Requirements

Currently, some solutions are proposed as means to address these problems in developing vehicle electronic systems. Here, we describe them in the light of these requirements.

Model-based development tools - Using a model to construct a system is always preferred to prototyping and testing due to cost and development time. The aim with using a model is to predict aspects of the system before constructed. Models of computer systems are currently not as mature as models of mechanical systems and the potential of achieving mature models is considered huge. Thus, the desired models should offer a high level view of the system allowing predictions on properties such as reliability, overview of system functionality and implementation. All with the aim at leveraging complexity – increasing quality and reducing cost.

Current model based tools include code generators in such tools as Rational Rose [5] and Rhapsody [6], where graphical representations of a software system automatically generates implementation.

The unified modeling language, UML [7], is intended to provide such a high level model where the system can be described using object oriented graphical notations. UML also include use case diagrams which can be used for specifying system functionality.

The goal of modeling clearly addresses requirements on cost, reliability, and integration. As models become more mature, OEMs of automotive vehicles can reduce the number of prototypes during development.

Software architecture - As computer hardware is getting cheaper while housings, connectors, and cables are not, we will get more processing power and reduce product cost mainly by reducing the number of control units. Fewer control units implies more software in each one. OEMs that come up with methods to integrate software components from different vendors in the same ECU will, thus, be able to reduce product cost. Challenges in achieving this goal include problems with specification, intellectual property (IP) issues, safety, and verification.

To make this feasible, software architectures are investigated that provide the necessary mechanisms for automotive applications and at the same time can be agreed upon by many OEMs and suppliers making it a standard.

The EAST-EEA project [8] involved some of the European OEMs and first tier suppliers, and investigates unifying the run-time environment (and also development process) for on-board software. One goal of this project is to define a software middleware based on OSEK specifications in order to allow integration and partitioning of software components. The AUTOSAR partnership [9] of European OEMs and tier 1 suppliers, have a similar objective and will develop and try to establish an open standard for automotive software architecture.

Network technologies - In order to meet requirements on safety, network technologies such as Flexray [11], TTP/C [12], and TTCAN [13] are proposed. These technologies include bounded message delay, global clock, and fault tolerance. These mechanisms all aim at assuring function and providing a more reliable communication link that provides means to ensure safety related transmissions. These busses are all based on the time-triggered paradigm where the progression of time initiates data transfers rater than asynchronous events. The time-triggered busses provide synchronous communication without the need for arbitration. Therefore the time-triggered protocols are suitable for implementing safety critical control functions with stringent demands on low latency and low jitter.

Low cost busses have been introduced in automotive applications in order to facilitate cost effective integration of components such as smart sensors and actuators into the vehicle network. Smart sensors and actuators have some ability to process (typically filter, or translate) measurements and send signals on the network whereas non-intelligent ones are wired to the I/O of an ECU that handles processing. The introduction of low cost controllers and single-wire networks is made at the expense of bandwidth, which is relatively low for these busses. The low cost busses also present a way of reducing complexity of the master node and facilitate variants in differently equipped products with only one ECU configuration.

Since vehicles are becoming equipped with more and more multimedia and telematics applications, the need for dedicated infotainment busses has arisen. A network in this category is MOST (Media Oriented Systems Transport) [3], which is based on optical fibre technology, and provides high bandwidth and services optimized for infotainment applications.

By wire solutions - Inside the computer system, everything can be considered to be a “by-wire” solution, but generally exchanging crucial functions like steering and braking is considered when using the term by-wire. As reported in this study, many functions are already implemented using the computer system. However, computer control of all the crucial functions to do with maneuvering the vehicle is considered as a shift in paradigm. In order to do so, the OEM must be confident that the computer system is at least as safe as a passive system and this is shown to be more difficult in computer systems as the failure modes increase [14]. The systems that are considered to have a safe state, e.g. the throttle, are easier to change into by-wire and all the vehicles investigated in this study have by-wire accelerator.

The trend towards by wire solutions is strong because of the envisioned benefits. Decreased product cost and numerous new types of functions can be offered. The product cost would become reduced because of removed hydraulic and mechanical links. Also many new functions would be facilitated, many of which are safety enhancing functions, such as emergency braking and collision avoidance. The overall layout of the vehicle would also become more flexible as fixed mechanical solutions are removed.

4. CONCLUSION

We have presented four case studies of vehicle electronic architectures in their business situation; in this describing the business context, business requirements, and resulting electronic architectures.

We have shown that challenges in cost, integration, variants, brands, and commonality as well as challenges in functionality, aftermarket, and safety are important to OEMs design decisions. Further, there are parameters in the business context of an OEM that strongly affects design decisions such as product volume and size of the market. The analysis shows that despite a common base of similar vehicle functionality the resulting electronic architectures used by the four organizations are quite different. The reason for this becomes apparent when looking at different business context and business requirements and their affect on the architecture. Differences in business situation drive the use of different methods for integration, different standards, different number of configurations, and different focus in the development effort. Some key parameters in business situation affecting architectural design decisions are shown to be product volume, size of market, and business requirements on openness and customer adaptation.

An important lesson from this is that one should be very careful to uncritically apply technical solutions from one industry in another, even when they are as closely related as the applications described in this work. Understanding the requirements from the business case is the key to choosing architectural solutions.

Against the background of this study, we have also reflected on some contemporary trends in automotive electronics today and provided discussion topics and speculation on why certain solutions are in focus today. These speculative sections include the topics; model based development tools, software architecture, network technologies, and by-wire solutions.

5. REFERENCES

[1] Fröberg, J., Engineering of Vehicle Electronic Systems: Requirements Reflected in Architecture. Technical report, Technology Licentiate Thesis, No.26, ISSN 1651-9256, ISBN 91-88834-41-7, Mälardalen University, March 2004.

[2] Specks, J., Rajnák, A., LIN - Protocol, Development Tools, and Software Interfaces for Local Interconnect Networks in Vehicles,

9th International Conference on Electronic Systems for Vehicles, Baden-Baden, October 2000.

[3] MOST Specification Rev 2.3, MOST Cooperation, www.mostnet.de, 2004.

[4] SAE Standard, SAE J1587, Joint SAE/TMC Electronic Data Interchange Between Microcomputer Systems In Heavy-Duty Vehicle Applications, www.sae.org.

[5] IBM, Rational software, http://www.rational.com.

[6] I-Logix, Rhapsody, http://www.ilogix.com.

[7] UML, Unified modeling language, http://www.uml.org.

[8] EAST-EEA, ITEA-Project-No. 00009, http://www.east-eea.net/

[9] Autosar partnership, Automotive Open System Architecture (AUTOSAR), http://www.autosar.org/

[10] A. Möller, J. Fröberg, M. Nolin; What are the needs for components in vehicular systems? - An industrial perspective -, In Proceedings of the WiP Session of the 15th Euromicro Conference on Real-Time Systems, Porto, July, 2003.

[11] FlexRay Requirements Specification, Version 2.0.2 / April, 2002, www.flexray-group.com.

[12] Time-Triggered Protocol TTP/C, High-Level Specification Document, Specification edition 1.0.0 of 4-July-2002, TTTech Computertechnik AG.

[13] Road Vehicles - Controller Area Network (CAN) – Part 4: Time-Triggered Communication. International Standards Organisation (ISO). ISO Standard-11898-4, December 2000.

[14] Czerny, B.J.; D'Ambrosio, J.G.; Murray, B.T., Providing convincing evidence of safety in X-by-wire automotive systems, High Assurance Systems Engineering, HASE 2000, Page(s): 189 - 192.

Paper B

Key Factors for Achieving Project Success in Integration of Automotive Mechatronics

Joakim Fröberg Mikael Åkerholm Kristian Sandström
Christer Norström

Published in the Journal of Systems and Software Engineering,
April 2007, DOI 10.1007/s11334-007-0023-2

Abstract

In this paper, we present a multiple case study on integration of automotive mechatronic components. Based on the findings, we identify that the root causes of problems in integration are largely related to decisions omitted in electronic strategy. We present and recommend use of checklists defining key factors to address in order to achieve successful integration projects in terms of cost and quality. Our recommendations are defined by checklists for critical decisions in areas; functionality, platform, integration, and stakeholder involvement. The recommendations are established based on practitioner experience and then validated in a multiple case study. Five cases of integration are studied for different heavy vehicles in one company, and the fulfillment of our recommendations is measured. Finally we define project success criteria and we compare the level of fulfillment with the project success in terms of time plan and resource consumption. The main contribution of this study is the validated recommendations, each including a set of checkpoints that defines recommendation fulfillment. We also present defining characteristics to identify a high risk project. We provide a set of observable project properties and show how they affect project risk.

1 Introduction

The majority of functions in a modern vehicle are partly controlled by electronics, i.e., software controlling physical devices via electronic hardware. As the electronic system becomes more defining for vehicle behavior, the integration focuses more and more on the electronics. "*Ninety percent of innovations in a modern car are based on new developments in electronics*" [22] is stated by one of the worldŠs largest suppliers of automotive components. To fully benefit from these innovations and in order to achieve the valued qualities of any vehicle such as comfort, energy optimization or performance, the integration of electronic systems plays a vital role.

Demands for functionality in a modern vehicle together with the market availability of electronically controlled mechatronics yield a situation where the automotive Original Equipment Manufacturers, (OEMs), design products by integration of subsystems. The behavior and qualities of the vehicle are much dependent on the electronic control of physical components and, often also, on the close co-operation of different electronic vehicle functions. As the complexity of modern in-vehicle electronic systems increases and imbues all vital components, the integration effort has a strong focus on electronics. Also, as complexity grows the integration of electronic systems has proven increasingly difficult and automotive OEMs find cost and quality estimates challenging. OEMs of automotive products want leverage over targeted qualities and, at the same time, the cost of scale when purchasing supplier components.

1.1 Automotive Integration

Original Equipment Manufacturers (OEMs) of automotive vehicles face a business situation where a product consists of numerous components; and where the components originate both from internal and external suppliers. Components from external suppliers are typically used wherever development cost and project risks are deemed beneficial compared to arranging internal development. Thus, one task of the OEM is to integrate components to form an overall system design that constitutes a vehicle. Many of the components available in the market of automotive components are mechatronic, i.e., besides the mechanical parts they include embedded electronics. Examples are brake-, engine-, hydraulic-, and climate-systems, all which typically include advanced electronic systems. These electronic systems need to interact with other in-vehicle systems to deliver the intended functions. An example is an Electronic stabilizer program, ESP, where braking, engine, and suspension systems collaborate to achieve its function. In-vehicle computer system design is therefore partly done by designing integration solutions. The overall goal of the electronic system design is to achieve a system that delivers its function with targeted qualities and is feasible to produce and service. Desired qualities such as reliability, safety, and maintainability affect choices in platform architecture. For instance, to achieve high reliability and enable safety analysis OEMs often use buses and protocols with fault tolerance and bounded transmission time. The need for maintainability drives architectural choices in diagnostic systems such as standardized ways of signaling faults. Cost targets drive the use of platforms both for the complete vehicle as well as for the in-vehicle computer system. An in-vehicle computer platform is a set of design decisions, components, processes and tools that is reused between vehicles [8]. The architectural choices related to the electronic system are manifested in the platform. Examples are operating systems, communication buses, software component models, but also design principles such as a principle of allowing only cyclical messages on some critical bus. A platform has longer life span than a single product and its design is not freely changed during vehicle projects. Choices in diagnostic strategy and fault handling, for instance, are not made for each vehicle and often cannot be altered during integration of a component. A supplier of an electronic component designs the component with desired qualities and cost targets and makes different architectural choices. There is a possible architectural mismatch and the electronic component can conform more or less well to its intended environment.

Thus, when integrating a component in an existing platform we are presented with design constraints both from the platform and the component. In order to find a design that meets all requirements and constraints, an integration solution is desired. Here, we refer to the process of doing this design as integration. Given an off-the-shelf component and a platform with largely decided architecture, an integration project can involve redesign or design of an adaptation. Thus, in order to achieve an integration solution we have the following parameters to change; 1, Revise the component, 2, Revise the platform, or 3, design a "glue" solution, indicated by the dark adaptation area in Figure 1. An adaptation is anything that is required to get the intended functionality and quality from the component. Examples could be software to translate signals into a desired format, adding memory

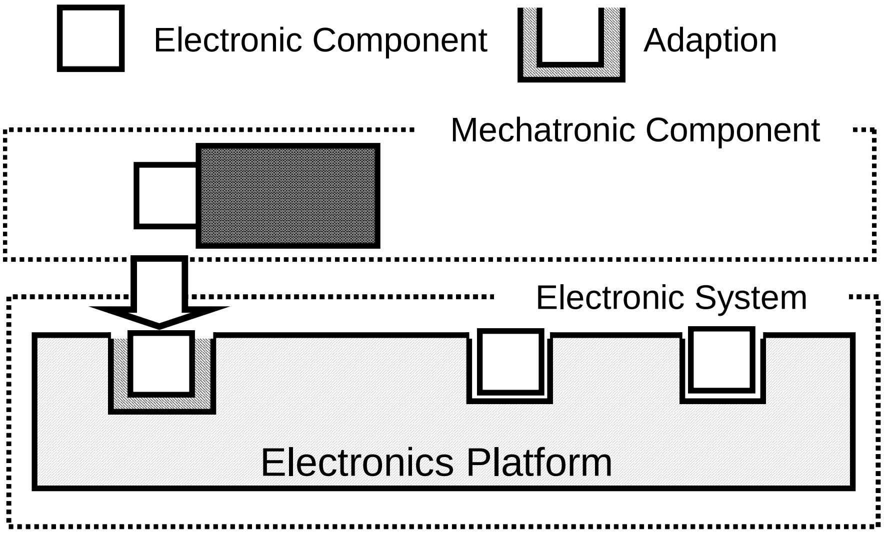

Figure 1: In-vehicle electronic system design by integration

protection, adding a bus gateway, changing of I/O, or freeing resources to satisfy the component. In an automotive context, the revision of a component can typically include a changed interface for the services provided of the component such as diagnostic and fault reporting. Also the functionality of the component can be extended to support different modes of operation, e.g., energy, or safe modes, or to support better control capabilities.

1.2 Problem

Modern vehicles contain electronics in all vital components and the success of a vehicle is dependent on the in-vehicle electronic system. OEMs experience a larger portion of the vehicle development projects are spent on electronic systems [14]. Development of electronic systems is typically performed late, close to production start, and is therefore critical to meet plans for production. The effort of integrating electronic systems has proven difficult with respect to assessing project success in terms of time and cost [3]. Automotive OEMs desire both the benefits in cost and functionality by using specialized suppliers, and an electronic system that enables successful vehicles: both in terms of vehicle behavior and life cycle support such as service and production. This puts focus on the OEM ability to integrate electronic systems.

An OEM used to develop computers and software in-house need to shift to a model of development more focused on system integration. Technical and architectural solutions for integration need to be investigated with respect to success of integration projects. Also, the engineering methods of integration need to be decided. To enable reasonable efforts in evaluating integration solutions, one key is to enable approximate evaluation with less than full information, and thus find key factors and simplified models [5].

1.3 Study Objectives

Ultimately, OEMs would want to have predictions on which platform architecture, integration solutions, and integration methods that lead to successful projects. In this study we have studied what practices and techniques that affect the outcome of integration projects and the goal of the study is to show which factors are the most important. The idea is to provide, not only, a list of affecting factors, but a list of key factors that should not be omitted when executing a project of mechatronic integration. The objectives of this study are:

1. To identify key factors that affect project success.
2. To propose a checklist with recommendations.
3. To validate the recommendations.

In this study we focus on the OEM problems in integration. By analyzing the cases we identify key factors and how they affect the outcome. We collect these factors and provide a checklist of best practices for an OEM of automotive products. Having the checklist, we aim to validate its recommendations by measuring fulfillment and project success in a series of projects. The overall objective is, thus, to provide a usable checklist for integration projects that includes key factors that are shown to avoid problems.

2 Related Work

As presented in the preceding sections, integration in the automotive domain is the effort on joining together mechatronic sub-systems from many suppliers. To achieve success in such projects, use of sound systems-engineering principles is a key to success. To assess a company's systems engineering capabilities the SE-CMM method can be used [9], the practices described in the method is sound and can also serve as guidance when developing processes and guidelines. SE-CMM practice area PA 05, treats our area, system integration. In our work we focus on describing the details of practical cases and assess which practices are the key factors in achieving success. In a study by Nellore and Balachandra [15], the systems engineering practices at a major European premium car manufacturer were reported. The study covers the entire project phase, and as a consequence, is not very detailed regarding sub-system integration. However, Nellore and Balachandras identifies that suppliers require specifications according to their history of OEM cooperation, and this is confirmed in our study. We focus on integration of the electronic part, i.e., software and electronics hardware, of the mechatronic component, because this part is often problematic to integrate. Thus, this is an area where more research is needed. There are two major design approaches to integrate the electronic part of a mechatronic component into the electronic system; we will examine both alternatives here:

- Hardware integration, which traditionally is the most common approach in the automotive domain. The integration here is realized through connecting a computer system, i.e., Electronic Control Unit (ECU), to a computer network in the vehicle. The integration interface is the computer network, and interaction with other parts of the system is through message exchange over the network.

- Software integration, which has much focus in current research and standardization efforts in the domain. In this case the integration is performed through deploying a software component on an ECU which is not part of the mechatronic component. The intention is to deploy several software components on the same electronic hardware; one goal is to decrease the number of ECUs. The software integration solution offer to partition software and configure ECUs more freely. In this case the integration interface is the software environment on the receiving ECU.

To simplify the specification and integration work with hardware components, standards can be used. Common in vehicle industry today is to use CAN [11] busses as physical medium with standard protocols on top, e.g., SAE J1939 [23] specifying the syntax and the sematics of certain messages including message identifiers and value range, as well as their meaning. In terms of integration this gives the OEM and suppliers a common agreement on bus interface. Another common method in combination with CAN is to use tools to package individual signals into proprietary messages on the bus and derive priorities so that temporal constraints are met, e.g., the Volcano tool [6]. In this case the OEM-supplier agreement is not standardized bus messages, but standardized communication software on ECUs which let the OEM configure bus traffic. Furthermore, there are numerous scheduling algorithms that can be applied on top of CAN providing bandwidth and timing guarantees for ECUs independent of the behavior of other ECUs. Typically these algorithms limits the effects of the node driven access to the bus through introducing time-driven access in some form, e.g., TT-CAN [12] and server CAN [19]. Nolte et al presents a survey of different alternatives that facilitates sub-system integration in the context of the CAN protocol in [18], including the discussed J1939, Volcano, TT-CAN, and server-CAN. There are also recent automotive communication protocols that are time-triggered in the basic specification, e.g., FlexRay [10]. Standards have also been used to simplify integration of software components. In the automotive domain OSEK/VDX [20] aims to define an open architecture that standardize software interface to communication, network management, and operating system. An ongoing standardization effort is the AUTOSAR [4] project, its goal is to create a global standard for basic software functions such as communications and diagnostics. From an integration point of view, AUTOSAR provides mechanisms for routing communications between software components regardless of their locations, both within a node and over networks. There are currently much research efforts in the area of software component technologies for embedded systems; several of these results could be interesting to support integration of software components in our context. However, availability of suppliers supporting the technologies is a requirement by OEMs before adopting it as integration platform. A major ongoing project is the DECOS project [7]. DECOS is focusing on safety critical systems, e.g., automotive and aerospace applications. The core of DECOS is a time-triggered architecture, providing both spatial and temporal partitioning, preventing interference among components sharing access to devices, as well as timing interference between components. Much of the current research work focus on smaller software components to be used for in-house development, e.g., Koala [24] and PECOS [17]. SaveCCT [1] is also one of these technologies, but in current efforts we try to expand the size of components to suit subsystems traded between suppliers and OEMs [2].

3 Method

This section describes the method used to perform the study. First, we state what basic steps are done during the execution of the study. Also, we present our way of measuring project success and our reasoning on validity of the results.

3.1 Method Overview

In the initial investigation, integration problems in three of the five cases were studied. The data was collected by in-depth interviews with engineers, project managers, and specialists. Based on the findings of this study, our analysis gives a set of measures that would counteract the reported problems. These measures are listed in a guideline. We categorize measures into four categories which we label recommendations one through four, R1-R4. In order to validate the recommendations, a second study was made. We listed observable checkpoints to inspect to which degree each item in the guideline was fulfilled. Also, we defined criteria for project success. We applied the method to all five integration projects. The data collection in this second phase was performed by interviews and follow ups were made with phone calls and e-mail.

3.2 Criteria for project success

As mentioned, the problems in integration of automotive mechatronics relate to achieving both quality and cost of the complete vehicle and integration is an important factor contributing to these goals. In order to measure project success we rely on measuring the fulfillment of project plan, project cost, and planned product cost. We do not measure explicitly the outcome in terms of quality. However, the achieved qualities like serviceability and reliability of the vehicle is largely decided early by selecting strategies for diagnostics, fault behavior and more. The desired quality is therefore achieved if the project is executed as planned. Many of these strategies are, once chosen, not negotiable. If a component were to fail in complying to a decided diagnostic signaling scheme for instance, the world wide service organization may not be able to handle this component, which certainly would prevent the vehicle from being produced at all. Instead projects are delayed or more costly than planned, but the decided functionality is achieved. The fulfillment of the quality-wise important functions of the electronic system are thus measured correctly by our definition of project success. However, quality flaws of the electronic system itself such as bugs and faulty connectors would not turn up in our measure, and would instead require studying operation of the system. In the studied cases, there was no concept decisions revoked related to the integration, but certainly not all met the project targets on resources and product cost. In summary we rely on our definition of project success to include indirectly a measure of quality.

3.3 Validity of results

The results and recommendations are derived from five cases from a single company. Although the stipulated problem of integration is the same in another company, the differing context can yield different importance to the problems found in this study. One factor that may have impact and likely to be different is the platform architecture. The problems in integration are general to automotive OEMs, and our recommendations are valid to tackle the stated problems. But the severity of each problem may well differ in a different context, and our recommendations, although we show them to be necessary, may not be enough to counteract problems. It is however likely that many of our recommendations are valid within many automotive companies, although dedicated studies have to be made to verify this.

4 Recommendations

As mentioned we have initially analyzed data from three integration projects and present a checklist with recommendations to achieve project success. We then validate our checklist by collecting data on practices for each case and correlate that to collected data on project success. Thus, the first task was to find the root problems from the collected data and to set up logical countermeasures. The countermeasures were collected in four checklists corresponding to four different areas of concern. Our main hypothesis was that fulfillment of the checklists will yield project success. In this section, we describe the reasoning to support our recommendations. Later sections describe the validation of our hypothesis, i.e., the validity of our recommendations in a series of projects.

From the interviews, reported problems were analyzed and experience from involved staff was collected. This knowledge has been analyzed and elaborated into a list of recommended practices for integration of electronic sub-systems. Each recommendation includes several checkpoints that stipulate a strategy decision. We see the problems largely originate from early phases of decision making. The root causes of the problems come from failures to address choices in design strategy. Each choice in design strategy is here annoted by a checkpoint. We see from the study that omissions cause problems and consequently the recommendation is a set if checkpoints that should not be omitted. We support each recommendation with reasoning and findings from the study. Each checkpoint is shown as a leaf in the following figures, and each checkpoint represents a strategy that should be decided in order to comply with that recommendation.

4.1 Recommendation 1 - Functionality

Here, we present a detailed list of checkpoints that supports deciding on functionality. The checkpoints define recommendation R1. *Recommendation 1 -*

All the functionality of the component should be decided prior to designing integration solution. The study shows that problems arise when key areas of

functionality is not decided. Especially, the study shows that much of the focus prior to choosing component is on the operational functionality of the component while diagnostic functions and system interaction issues are omitted. Examples are system degradation behavior, fault signaling, and production tests, all of which often constitute a major part of the electronic system. Another typical problem reported was that the detailed technical issues of protocols, interfaces, and tools were wrongly estimated to be adaptable. We draw the conclusion that the system level functionality and all interaction between component and system should be decided prior to technical design. Figure 2 defines a set of checkpoints to counteract the reported problems. Each checkpoint is represented by a leaf in the tree and it corresponds to a decision that should be performed to comply with the recommendation. Decisions on timing, diagnostic functions, operation, and fault behavior were reported as being incomplete and to cause problems in some cases. Therefore

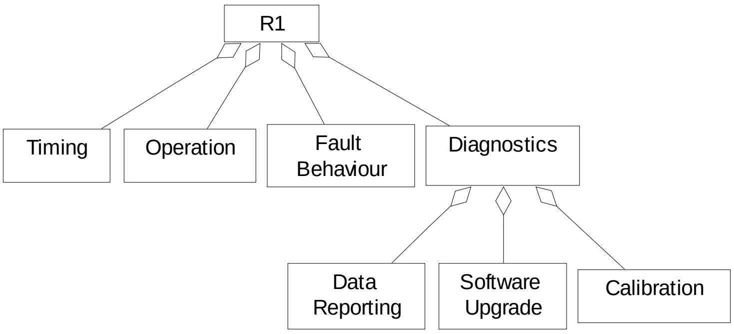

Figure 2: Checkpoints for Recommendation 1

we collect these decisions for the recommendation 1 checklist. First, decisions on timing include control parameters such as latency, period time and jitter. The diagnostic functions were reported in more detail and include general data reporting, software upgrade, and calibration functions. Data reporting functions are functions that report measurements, faults and status of the subsystem, e.g., sensor value and status. Software upgrade is the functionality that allows downloading new software in an ECU after the product is sold. This is reported important as it enables updates without replacing the physical ECU. In addition, many mechatronic components require calibration to compensate for component variations and this functionality has to be included and used in production. The operation checkpoint refers to the main function of the mechatronic component such as providing climate control. The last checkpoint in Figure 2 is fault behavior and this means typically the functionality of acting in accordance with a system wide fault state. The sematics or behaviour in each state should be decided for the given component. Also, the component in itself can introduce new fault states. All these checkpoints on functionality require decisions. To comply fully with the recommendation R1, all the functionality involved in each checkpoint should be decided.

4.2 Recommendation 2 - Platform

In Figure 3, we present a list of checkpoints that support decision making for what platform constraints that applies to this particular component.

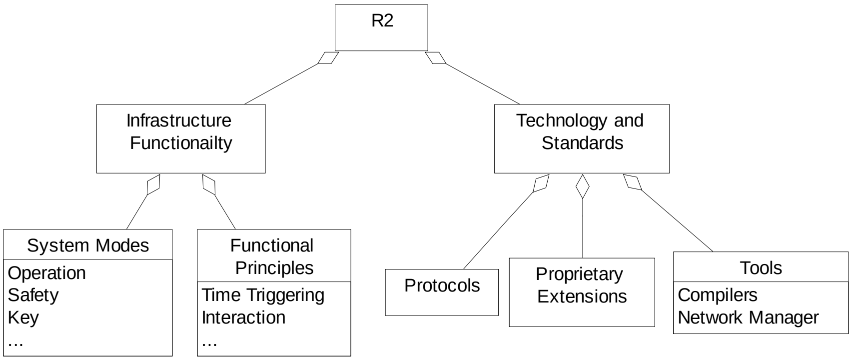

Figure 3: Checkpoints for Recommendation 2

The checkpoints define recommendation R2. *Recommendation 2 - Know the design constraints imposed by the platform prior to designing integration solution.* Decisions on functionality will act as implementation requirements, but in addition, there are requirements that originate from design decisions taken for the platform. An OEM vehicle platform has longer life span that the products it enables and its design is not normally changed due to the needs of a single integration or product project. Thus, the platform imposes constraints on how the component can be integrated. More precisely, the platform defines how components interact by its inherent choices in paradigm, technology, and infrastructure. This is true both for software and hardware components. The study shows that it is crucial to know these constraints to avoid project failure. Each checkpoint, thus, involves knowing one or more constraints and deciding to adhere to it. The critical decisions to take according to the study results are shown in Figure 3. The checkpoints are divided into constraints related to the infrastructure of the system and constraints related to choices in technology and standards. The infrastructure of an automotive electronic platform does include some mechanism to support different system modes and also it may involve functional principles or inherent system philosophies. The platform have explicit system modes such as safe mode, key modes, and perhaps other operational modes, and the component must provide functionality to support this. One example is a gearbox that could be made to reduce operation if some other part of the system experiences a critical fault and enters a limp home mode, where the vehicle is reduced to using only the first gear. Also, the same gearbox is perhaps to prevent gear shifting if the key is not turned on or to support energy efficient modes of operation. It must be known what system modes in the platform that is relevant to the component to be integrated to fulfill the checkpoint. The platform can also contain other principles of operation. System design principles can include paradigms such as time triggered software execution or bus communication, or a client server architecture in software. In the category of technology and standard restrictions, communication protocols are mentioned together with company proprietary extensions. Standards and OEM extensions stipulate syntax and semantics of messages on a communication bus and therefore limit the design space for integration. Interview data also show that tool dependencies have been unclear and supposedly caused problems in integration. Lacking knowledge and decisions on these issues are potential causes of problems. It can be argued that simply knowing the constraint does not automatically cause a correct design. A team of engineers could still choose to connect something that only fulfils basic structural constraints such as protocol syntax and physical bus connector. What the study shows, however, is that it is the knowledge does cause a correct design. Problems occur when a component is selected without knowing or considering platform constraints in detail. The checkpoints related to platform constraints are collected in recommendation R2.

4.3 Recommendation 3 - Integration

In Figure 4, we present a list of checkpoints that supports deciding on an integration solution for one candidate component. The checkpoints define recommendation R3. *Recommendation 3 - The integration solutions should be investigated and a strategy chosen prior to choosing component.* Data from the cases show that components have been chosen in early phases of concept design where both functionality and integration feasibility have been estimated. In order to evaluate one mechatronic component, we must consider both the component itself and the adaptation to the platform as indicated by the dark area in Figure 1.

Failing to address the adaptation, we will not know what component functions and properties that will become useful to the system. Our recommendation involves evaluating each candidate to compare effort and value

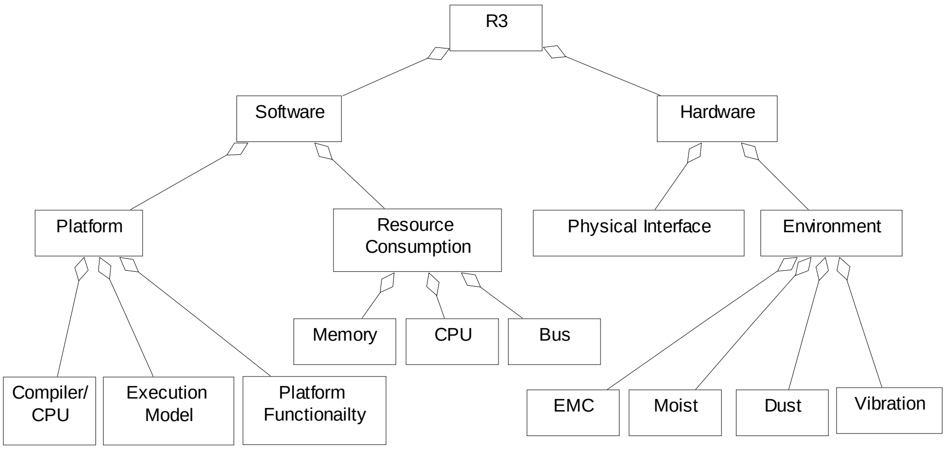

Figure 4: Checkpoints for Recommendation 3

before choosing candidate. However the checklist can be used even if the selection is already done. Thus, failing to evaluate all candidate components, at least the feasibility of integration should be evaluated for the chosen component. Not deciding on integration strategy according to these checkpoints impact the project resource consumption very negatively according to our study. Seemingly minor issues such as a conflicting bus message id, has later proved to be problematic to change. Here, there are two basic choices in integration strategy as shown in Figure 4. Either the strategy is to integrate an ECU on communication busses in the system (hardware integration), or to integrate software functionality into an existing ECU (software integration). The checklist for actions is different in the two strategies. Basically, in order to select the optimal component, we suggest evaluating both strategies and compare the effort needed given the wanted functionality. However, if there are reasons why the strategy cannot be freely chosen, the checklist can be applied for only the selected strategy, i.e., hardware or software integration. For hardware integration, the checklist includes decisions to make for physical interface and environmental requirements on physical parts. For instance, for a given functionality, the ECU may need to be connected to several networks and this should be explicitly decided and feasibility should be assessed. Also, there are decisions to make for environmental requirements. These are likely specified by standards and there may be different areas of

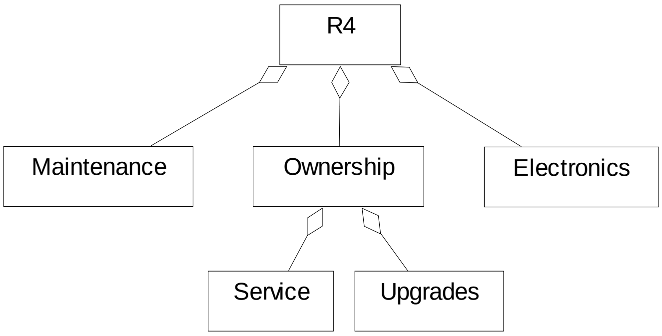

Figure 5: Checkpoints for Recommendation 4

the vehicle that implies different physical roughness. These decisions should be explicitly stated and agreed upon with suppliers. For software integration the focus is largely different. A software component can be integrated by deciding and specifying the software platform interface for the intended ECU host. Decisions should be made on compiler dependencies, execution model, and software platform services. Also, the resource consumption of a software component should be decided because there are limited system resources. There can be integration cases where hardware and software strategies are mixed, e.g., if a software component, an ECU, and a set of electronic sensors are delivered by different suppliers. The recommendation is still valid, but the internal software design issues become the concern of a supplier. All the checkpoints related to designing the integration solution, we have collected into the recommendation R3.

4.4 Recommendation 4 - Involvement and responsibility

In Figure 5, we present a detailed list of checkpoints to observe in order to address involvement and responsibility assigning. The checkpoints define recommendation R4. *Recommendation 4 - All stakeholders should be involved and the responsibilities should be assigned for the activities of the subsystem lifecycle.* The investigated cases show incompleteness in responsibilities as one likely reason for delay and increased project cost. There were several departments within the OEM that initiated projects involving electronics.

Also the electronic system spans most of the vehicle subsystems and it was not always decided what role was to be responsible for each electronic subsystem. Reportedly, roles in service, maintenance and electronics were not fully decided. Also ownership of designs was mentioned as a potential pitfall for the project outcome.

5 Five Cases Analyzed

Here, we present the data from the five studied cases of automotive mechatronic integration (section 5.1). We present case descriptions to show the context of each case. Also, data on fulfillment (section 5.2) and project success (section 5.3) is shown for each case. Any references to actual products or projects have been removed.

5.1 Case data

In Figure 6 an overview of the contents in the different cases is presented. The figure shows that all cases included the elements of software and mechanics, while whether electronics and ECU was included in the integration projects varied in the different cases.

Case #1 This project introduced computer controlled mechanics related to the drive train. A supplier offered a system with mechanical components as well as control system including sensors, actuator, computer hardware and software. The decision was made to purchase the mechanical parts with fitted sensors and actuators and the software as a binary component, but not the computer hardware. Thus, the algorithms controlling the mechanical parts are implemented in a software component by the supplier, which is integrated

Case #1	**Software**	~~**ECU**~~	~~**Electronics**~~	**Mechanics**
Case #2	**Software**	~~**ECU**~~	**Electronics**	**Mechanics**
Case #3	**Software**	**ECU**	**Electronics**	**Mechanics**
Case #4	**Software**	**ECU**	**Electronics**	**Mechanics**
Case #5	**Software**	~~**ECU**~~	**Electronics**	**Mechanics**

Figure 6: Case characteristics

into an existing ECU with a software platform owned by the OEM. The software component was originally developed by the supplier for another CPU with another compiler. Moreover, the source code was owned by the supplier and not to be made revealed to the OEM. The software component provided functionality that was central to the product in that it controls functionality in the drive train. The affected functionality has some safety implications due to the influence on vehicle handling. Initially the quality of the functional specification was poor and had to be redone during the project. Although this integration solution did not directly affect any physical design such as bus topology, the component impacts the software by making analysis and verification more difficult.

Case #2 This project developed a modular solution to provide a climate control in the cabins of construction equipment vehicles. Modules include; software component encapsulating climate control algorithms and a numerical keyboard with a communication bus interface. The computer hardware was an ECU provided by the OEM and contains a software platform with operating system and communication software components. Different sets of modules could be used in different machines and the solution is intended for integration in one of several ways, e.g., standalone, one bus connected, or with two busses connected. In the investigated case the solution was to have only the diagnostic bus connected. In this case there was at an early stage an overview specification on how integration was to be made with respect to communication, i.e., it was specified to adhere to OEM internal diagnostics protocol. The overall impact on the in-vehicle computer system was low in integrating this ECU. There were no safety implications and the climate control system is not tightly connected to the rest of the machine functionality. Only the diagnostic bus was to be connected and not the more critical control bus. In terms of maintenance the solution supports design change and replacements of physical components and software as well as would an internally developed system. The supplier of algorithms in this case was a company within the Volvo group. This supplier has more experience with Volvo specific requirements on diagnostics and general architecture than would a random automotive supplier.

Case #3 The objective of this project was to integrate a computer controlled hydraulic component to achieve a hydraulic function in a construction equipment vehicle. The embedded computer system consisted of an ECU with a control application and one CAN interface. Also included was a sensor with a CAN interface. This case shows safety implications and the functionality is central to the behavior of the product. The safety implications yield high requirements on ability to perform analysis and this, in turn, make integration more difficult. This component required a high degree of interaction with the vehicle electronic system. Many problems had to be handled during the project. The component did not conform to the present platform diagnostic system. Thus, an integration solution that translated diagnostic information was required. The fault behavior of the ECU was not specified at the start of the project nor was the bus communication. As a result, the ECU software needed late changes.

Case #4 In this case a project was run to integrate functionality related to the powertrain of a construction equipment vehicle. Decisions were made to purchase a complete system with mechanics, electronic hardware and software. The system that constitutes the component for integration in this case included a single CAN interface. The component has impact on products behavior and some safety implications. There were early decisions to avoid adopting the component functionality to platform diagnostic principles, e.g., software upgrade of this ECU was not to be supported. The problems encountered in this project were mainly related to the environmental requirements of the electronic hardware included.

Case #5 This case consists of a project to integrate a mechatronic component used for hydraulic control. The component consists of hydraulic components, electronic hardware, and software. Like in case #2, the software was decided to run on an ECU from the OEM. The component is central to the vehicles core functions and behavior, and it is safety related. The electronics of the component interacts with the in-vehicle system to a large extent, and thus its integration has high impact on the electronic platform.

5.2 Fulfillment of Recommendations

In order to measure how each of the investigated projects fulfill the recommendations, we have collected data on how the projects were run. For each checkpoint in the recommendations, we investigate if the corresponding decision was taken at the time of choosing component. Also, for each checkpoint, we determine if the decision was changed during the project.

R1 - Functionality The first recommendation R1 stipulates that decisions on functionality should be made prior to designing the integration solution. In Table 1, we present the decision status with respect to functionality for each project at the time when the component was chosen. The definition of the scale is shown in Table 2. Case #2 is shown to have the highest fulfillment and case #3 the lowest. There seems to be no strong correlation between the different types of functionality decisions; a component can have a poor degree of fulfillment in operation while having good fulfillment in fault behavior like case #5. One common response from the respondents of the interview was that early focus was aimed at only the operational functionality of the component while diagnostics and fault behavior was forgotten. There seems to be no support for this statement. Another explanation can be that this type of omission is done consistently throughout the projects and responses are relative to the "usual" poor decision making in diagnostics and fault behavior.

Table 1: Early electronics functionality decisions

	Operation	Diagnostics	Fault behavior	Timing	Overall fulfillment R1
#1	Good	Good	Good	Good	4.0
#2	Very good	good	Ok	Good	4.2
#3	Ok	Poor	Poor	Poor	2.2
#4	Ok	Good	Good	Ok	3.2
#5	Poor	Ok	Good	Good	3.2

Table 2: Legend for fulfillment of the functionality recommendation

5	Very good	Decided and complete specification
4	Good	Decided
3	Ok	Most decided
2	Poor	Little decided
1	Very poor	Nothing decided

R2 - Platform The second recommendation, R2, stipulated that the platform requirements should be known prior to designing the integration solution. We show project status with respect to the degree of decisions that were made on platform requirements at the time when the component were chosen.

The scale of measurement for fulfillment of platform requirement decisions is shown in Table 3. The first measure here is an average estimate by the involved people. The two following are measures of actual practices although thay show little span. Either we could rely on the estimates, the actual measures, or a combination. It seems the overall fulfilment would be in tha same range in either way and we conclude that the overall fulfilment measure of R2 as shown in Table 4 can be used to analyze the cases.

R3 - Integration Solution The third recommendation states that the integration solution should be investigated at least for the component to be integrated prior to running the project. In the studied cases we have collected data on both the degree of design decisions and the degree of deviation from these decisions. These measures are shown in Table 5. The scale of measurement for fulfillment of platform requirement decisions is shown in Table 6. As shown, case #3 and #4 have only three measures as software integration was not part of the integration and is not applicable. The average is thus calculated based on the three measures. Using the average value of the four measures could be misleading. If the major part of the integration was to integrate software, it seems logical that the third and fourth measures are

Table 3: Legend for measurements

5	Very good	Fully decided and no unexpected constraints revealed
4	Good	Largely decided and minor constraints were revealed
3	Ok	Few unexpected constraints were revealed
2	Poor	Unexpected constraints were revealed
1	Very poor	Unexpected constraint of major importance were revealed

Table 4: Platform requirements decisions

	Overall estimate by involved	System mode interaction	Dependencies to standards, technology, and tools	Overall fulfilment R2
#1	Good	Good	Ok	3.7
#2	Very good	Good	Good	4.3
#3	Poor	Poor	Poor	2.0
#4	Ok	Good	Poor	3.0
#5	Very good	Good	Good	4.3

Table 5: Integration solution

	Environmental requirements	Physical connection	Software platform	Resource consumption	Overall R3
#1	Very good	Very good	Good	Good	4.5
#2	Very good	Very good	Good	Good	4.5
#3	Poor	Poor	N/A	Very poor	1.7
#4	Ok	Very good	N/A	Ok	3.7
#5	Very good	Very good	Very good	Good	4.75

more important since they relate to the software integration strategy while the first and second measures are related to physical integration. However,

Table 6: Legend for measurements

5	Very good	Fully decided and no unexpected constraints revealed
4	Good	Largely decided and minor constraints were revealed
3	Ok	Few unexpected constraints were revealed
2	Poor	Unexpected constraints were revealed
1	Very poor	Unexpected constraint of major importance were revealed

as we can see in Table 5, all the projects show some elements of software integration decisions. Only when an ECU with very little changes is integrated, do the software platform decisions become fully the issue of the supplier, like case #3 and case #4. We use the average as calculated in the table, but remember that these represent two different sets of decisions.

R4 - Involvement and Responsibilities The fourth recommendation stipulates that all stakeholders should be involved and that their responsibilities should be decided. We have elaborated and collected data as to show to what degree this was done prior to running the projects. In Table 7, we show the degree of decisions combined with the degree of changes during the project. The definition of the scale is shown in Table 8, and the range is selected to include the data span from the study. The first measure of early stakeholder involvement represent how many stakeholders were involved early in relation to how many were involved in the end. The electronics people are one stakeholder, and thus this measure is part of the first measure. The respondents stated that late involvement of the electronics department is a problem and it seems logical to assume that this is an important stakeholder.

Table 7: Early stakeholder involvement

	Early involvement of stakeholders	Electronics involved early	Responsibilities assigned	Overall fulfillment R4
#1	Good	Ok	Good	3.7
#2	Very good	Good	Poor	3.7
#3	Poor	Poor	Poor	2.0
#4	Good	Ok	Good	3.7
#5	Good	Very good	Good	4.3

Table 8: Measurement legend for Early stakeholder involvement

5	Very good	Fully decided who
4	Good	Mostly decided who
3	Ok	Some decisions on who
2	Poor	Few decisions of who
1	Very poor	No decisions

Thus we combine the measurements for involvement and responsibility and use the average as an indication on fulfillment.

5.3 Project Success

In order to measure the success of each project, we have collected data on how the project was planned at the time of choosing the component. We use three measures and compare the initial plan with the actual outcome. We look at the projected time of completion, the projected product cost for the component, and the projected development cost. The comparison with the outcome is rated according to the legend shown in Table 9, and put into Table 10.

The definitions of the different levels of the Likert scale are shown in Table 10. The measurements in this case, represent the degree to which a plan was met. However, the interviews yielded explicit praise in two cases in terms of the project cost and we include the *slightly better than plan* measure accordingly.

The compilation of measures shows case #3 that stands out, where the overall result is especially poor. The others are in the range of Ok to Good. Case #2 shows the best measure of success. All five projects have no eye catching distribution in the different measures; all three measures seem to be coherent. The exception is case #5 where product cost is rated two levels higher than the other success measures. It is reasonable to believe the time plan and the development cost are highly interdependent, and that is also the case according to the data. The plan for product cost is also interrelated to the quantities of time plan and development cost, although it seems to a lesser extent. We draw the conclusion that the overall measure of project success can be used for analysis.

Table 9: Legend for Project Success measures

5	Very good	Plan met and involved personnel praise the outcome
4	Good	Plan met
3	Ok	Deviations less than 10%
2	Poor	Deviations less than 50%
1	Very poor	Deviations more than 50%

Table 10: Project Success

	Time plan	Product cost	Development cost	Total measure of success
#1	Good	Good	Good	4.0
#2	Good	Very good	Good	4.3
#3	Very poor	Poor	Very poor	1.3
#4	Ok	Ok	Ok	3.0
#5	Ok	Very good	Ok	3.7

6 Success Factors Analyzed

Previously we have shown the checklist for decisions followed by data on project success and the degree to which recommendations were followed in each of the five projects. In this section we analyze the correlation between checklist fulfillment and project success. Also we analyze the possible impact of other parameters that has been reported in the study.

6.1 Factors that Cause Success

For each recommendation, fulfillment was measured by a rating on the Likert scale, and each corresponds to a number one through five. The overall fulfillment indicator of each recommendation was then calculated as the average of the different measures. The same overall indicator was calculated for the project success measures. In Table 11, the overall fulfillment indicators for all four recommendations are listed together with the project success indicator. The total fulfillment of recommendations is calculated as the average of

Table 11: Project Success

	Overall R1	Overall R2	Overall R3	Overall R4	Total average	Total measure of success
#1	4.0	3.7	4.5	3.7	4.0	4.0
#2	4.2	4.3	4.5	3.7	4.2	4.3
#3	2.2	2.0	1.7	2.0	2.0	1.3
#4	3.2	3.0	3.7	3.7	3.4	3.0
#5	3.2	4.3	4.7	4.3	4.2	3.7

the four overall fulfillments. We see that there is a correlation between fulfillment of the recommendations and the achieved project success. Although the numbers are just indicative, the trend can clearly be seen that the recommendations do affect project outcome. Case #1 and #2 show high ratings on project success and also show high ratings on the fulfillment measures. Case #3 shows a poor fulfillment and the project success gets the lowest measure of project success. Case #4 shows a more moderate rating on fulfillment and a similar rating on project success. The last case, case #5, show a high fulfillment rate, but not a correspondingly high rating on project success. If we look at case #5, we see a project with high degree of fulfillment in all measures but R1, the functionality recommendation. So high that the average fulfillment is 4.2 in column five and still the total success is only 3.7. One explanation is that the decisions regarding functionality are more critical than others. Case #4 supports this explanation to a small extent. More precisely, if we look at the fulfillment of R1 in case #5, Table 11, we see that the operational functionality was decided poorly. We conclude that functionality and especially operational functionality could be critical to the overall project success. Intuitively this makes sense as a project with few functional decisions resembles an investigation more than a regular development project. Case #3 show the poorest success and also the poorest fulfillment of recommendations. Moreover it shows the largest difference between projected outcome according to column 5 of Table 11. However, analyzing the significance of this is difficult since this project is alone in that range of measurements. The correlation between projected outcome and actual project success is not necessarily expected to yield the exact same numbers, just as a trend. What this project does show us, is however, that the project success is indeed correlated to the fulfillment of our recommendations. In summary, based on these results, we conclude that the fulfillment of the four recommendations as stated is a prerequisite to achieve project success in an automotive integration project. Possibly, the early decisions on functionality are especially important to the project outcome.

6.2 Recommendations revisited

The data from the in-depth interviews attracted attention in two details. In the cases #1 and #4, the respondents state that a certain decision was made early. As we inspected documentation and outcome, it became clear that in case #4 the decision was in fact not taken and in case #1 it was seemingly a wrong decision. In case #4 all involved worked under the assumption that a physical property of the component was decided and specified, but the component did not in fact exhibit this property. This fact was discovered late and as a consequence late changes were required. In case #1 there was basically a good level of specifications early, but one decision was made seemingly under the wrong assumptions. All the decisions in our recommendation 3 were basically fulfilled, but also decided was that a software component was to be delivered as a binary compiled by the supplier to protect intellectual properties. This decision was made although the component was originally developed for another compiler. During the project there were three cases of bugs that could not have been solved if the supplier had kept the code hidden. Thus, in this case the problems were solved and accordingly this project does not suffer in project success. Since decisions have to be communicated and there is always a risk of misunderstandings and erroneous decisions, integration projects should involve re-assessments. Both these problems could possibly be avoided by matching the delivered component to the specifications early. Thus, we conclude that it is not always enough to make the decisions, but they must also be reviewed for misunderstandings and correctness. The studied cases would have benefited from a recommendation "Review decisions on integration and check that delivered components match decisions". Physical properties can be reviewed as soon as there is a component available, sometimes even at the time of choosing component.

6.3 Characteristics of High Risk Projects

The recommendations R1-R4 do assist in achieving project success, and thus a low fulfillment of our checklist will increase the risk of project delay and added costs. But were there other parameters in the context of the projects that affected the outcome? The study reveals some contextual parameters that could affect the level of risk. In order to analyze the impact of each factor, we present Table 12. It seems reasonable to check if especially difficult technical integrations have yielded low success projects. Therefore, we present data on the level of technical impact on the electronic platform, the level of safety related functionality, and the degree of impact on product behavior. These measures are listed in column one through three respectively in table 12. The degree of experience with the supplier has been reported an influential factor in integration [16] and thus we include this in the table.

Table 12: Project characteristics

	Integration impact on the electronic platform	Safety criticality	Impact on overall product behavior	Degree of experience with the supplier	Strategy, degree of software focus	Total fulfillment average	Total measure of success
#1	Medium	Medium	High	Medium	High	4.0	4.0
#2	Low	None	Low	High	Medium	4.2	4.3
#3	High	High	High	Low	Low	2.0	1.3
#4	Medium	Medium	High	Low	Low	3.4	3.0
#5	High	High	High	High	Medium	4.2	3.7

Also, interesting to inspect is the impact of the integration strategy; whether it is software integration or if the integration includes hardware and ECUs. Case #1 that largely consists of a software component gets a high rating.

There seems to be no conclusive indication that the decisions on hardware/software strategy cause differences in projected and actual outcome. Case #3 and #4 involves pure hardware integration strategies and they do show a small difference between projected and actual outcome. The low degree of supplier experience in cases #3 and #4 coincides with the strategy decisions and this factor could also explain the increased difficulty which in turn would impact outcome negatively. The importance of previous supplier experience was not particularly stressed during the interviews, but the correlation seem to exist. Case #5 and #3 stands out as they both show high ratings on the three first measures. All these measures supposedly give a more difficult integration and that seems a likely explanation for the difference between projected and actual outcome. At the same time case #5 and #2 show a high degree of experience with the supplier, a factor which would supposedly aid in achieving success. Case #2 show a low level of difficulties in the three first measures and at the same time a high level of experience with the supplier and that could explain that the actual outcome is actually higher than projected by the total fulfillment measure. The data in table 12, and our reasoning leads us to conclude that high risk projects are characterized by severe requirements on technically tight integration, safety, and a close relation to product behavior. The data is in line with this proposition and reasoning seems to suggest impact by constraining the integration. We also conclude that a low degree of experience with the supplier is a risk to project success. The data does not contradict this previously reported fact. Furthermore we conclude that the choice of strategy in integration does not have significant impact on the success. Hardware or software centered integration can be chosen on other premises.

6.4 Applicability of the Recommendations

We have presented a ready to use checklist outlining what decisions are important to make in order to reduce risk of project failure. Will we succeed if we follow the guideline in a given integration project? The study did likely not provide a complete picture of the factors that affect project success or failure. It is resonable to believe that issues of resources, competence, organization and more do also affect outcome. However, these issues were not reported as the main problems by the practitioners that were interviewed. Instead, they reported issues of functionality, platform, integration design, and responsibility. Therefor, we conclude that the recommendations are valid to tackle the stated problems in any context, but there still may be other factors that cause a project to fail. For the studied cases, other factors were of lower importance or at least not reported to be problematic. Thus, the recommendation should be applicable in any automotive integration project to reduce risks, but does not surely yield a successful project

7 Discussion

7.1 Implications to OEMs

An interesting implication from the integration recommendation, R3, is that the success of using a certain mechatronic component is not only dependent on the quality of the component, but also on the integration solution. Thus, it is wrong to assume a component such as an engine is the best choice based on success in another system without assessing integration with the intended system. Even if the component work flawlessly, and exhibits a series of valuable operational properties, an integration project could prove overly expensive, or worse, prove that the valuable properties are not achievable in the targeted platform. Another implication from the study is that mechatronic components should not be chosen solely by a domain related department such as a brake, or hydraulics department. The reason is that mechatronic components are per definition multi-domain components. To put the implication bluntly "An XYZ component should not be chosen by an X department" or the solution will be sub-optimal. For instance the engine department should not alone choose an engine even if they do have the best skills in engine operation and performance. Instead, a systems engineering principle of involving all stakeholders of the component lifecycle should be exercised. As the degree of electronics in modern mechatronic components is significant and contributes to all phases of the component life cycle, it is especially important to involve OEM electronics people to decide on component selection. Recommendation R3 implies that one should examine both the software and the hardware branch in order to choose what integration strategy is to be pursued. The cost and feasibility of each strategy should be compared and the choice made accordingly. If the strategy is set early by some reason, then we will not know which strategy was the cheapest and most feasible. Hence, we might end up with an overly expensive and difficult choice. It could be noted that an early decision on strategy before choosing component is non-optimal, since it is fully possible that it negatively affect project success. It would thus be wrong to choose a mechatronic component with included ECUs without considering the possibility of integrating the software on an OEM platform ECU. Also it would be wrong to assume that a supplier who does not know the platform constraints or the system goals, to recommend a solution. Most problems reported in the study indicate difficulties in achieving functionality, or quality for the whole electronic system such as constructing a maintainable, or fault tolerant system. These targets are achieved by choices in system strategy, e.g., diagnostic or software upgrade concept. Those concepts are not part of a supplier component. Thus, we conclude that problems in mechatronic integration do not stem from poor quality components or suppliers, but from the OEM.

7.2 Future studies

Our recommendations are validated by an in-depth, case-rich study, but some questions cannot be answered in a single company. In order to fully study the impact of the platform architectural decisions and answer questions like "How to design a platform for ultimate integration capabilities?", more companies could be studied. The same reasoning holds for organizational matters.

Making integration design is essentially the same as making architectural design, but with all the decisions previously made for the platform. Thus, the integration design is severely more constrained that a general architectural design. There are methods for architectural evaluation, where decisions are evaluated with respect to the targeted qualities, e.g., The Architectural Tradeoff Analysis Method (ATAM) [13]. In order to produce a design guideline for integration design, such methods could prove useful. The concept of making the right decisions is central to all design. There are methods to support in decision making, e.g., the Analytical Hierarchy Process (AHP) [21]. Such a method involves weighting of selection criteria and could also prove useful to support a future method for integration design.

8 Conclusion

We have presented an industrial multiple case study of integration of automotive mechatronic components. Based on the studies we have presented key factors for achieving project success in similar integration projects. We present six recommendations, the first four including detailed checklists for decision making. The recommendations are described by a brief summary of the checkpoints included under that topic.

1. All the functionality of the component should be decided prior to designing the integration solution; this includes diagnosis, production, and service functions.

2. Know the design constraints imposed by the platform prior to designing integration solution, e.g., global systems modes, communication protocols, and all constraining paradigms.

3. The integration solutions should be investigated and a strategy chosen prior to choosing component; this should include investigation of environmental requirements, and resource consumption.

4. All stakeholders should be involved and the responsibilities should be assigned for the activities of the subsystem lifecycle.

5. Review decisions on integration and check that delivered components match decisions as soon as possible, to detect misconceptions early.

6. Be aware that integration projects characterized by a technically tight integration, safety criticality, close relation to core vehicle behavior, or inexperienced suppliers are high-risk projects.

We show that early estimates of integration solutions are intrinsically difficult with less than designing every detail, but the checklist include key decisions that have been shown to counteract problems in real cases. The study shows that decisions can be erroneous or misunderstood, and we add the fifth recommendation to counteract misconceptions. We show analytically how our fifth recommendation would have solved misconception problems that did occur in the studied cases. Finally, recommendation six brings attention to what characterizes a high-risk integration project. We provide a set of observable project properties that can be used to identify high risk projects. The main contribution of this study is the recommendations each including a detailed set of checkpoints that pinpoint critical decision making and thus enables success in integration projects. In summary we conclude that successful integration of mechatronic components in automotive products relies heavily on decision making in electronic system strategies, and this study provides a detailed set of validated recommendations that assist in achieving just that.

References

[1] Mikael Åkerholm, Jan Carlson, Johan Fredriksson, Hans Hansson, John Håkansson, Anders Möller, Paul Pettersson, and Massimo Tivoli. The save approach to component-based development of vehicular systems. *Journal of Systems and Software*, May 2006.

[2] Mikael Åkerholm, Jan Carlson, Johan Fredriksson, Hans Hansson, Mikael Nolin, Thomas Nolte, John Håkansson, and Paul Pettersson. Handling subsystems using the savecomp component technology. In *Workshop on Models and Analysis for Automotive Systems (WMAAS'06) in conjunction with the 27th IEEE Real-Time Systems Symposium (RTSS'06)*, December 2006.

[3] Auer G. Automotive news europe, mercedes will make cars less complicated, technology leader cuts 600 electronic functions, seeks more, 2004. May 31, http://www.accessmylibrary.com/coms2/-summary_0286-21549689_ITM (Last Accessed: 2006-10-17).

[4] AUTOSAR. Homepage of automotive open system architecture (autosar), 2006. URL: https://www.decos.at/ (Last Accessed: 2006-01-17).

[5] Jakob Axelsson. Cost models with explicit uncertainties for electronic architecture trade-off and risk analysis. In *Proceedings of The International Council on Systems Engineering (INCOSE) 2006*, 2006.

[6] L. Casparsson, A. Rajnak, K.W. Tindell, and P. Malmberg. Volcano - a revolution in on-board communication. Technical report, Volvo Technology Report, 1998.

[7] DECOS. Dependable embedded components and systems, 2006. URL: https://www.decos.at/ (Last Accessed: 2006-01-17).

[8] Ulrik Eklund, Askerdahl, Granholm, and Jakob Axelsson. Experience of introducing reference architectures in the development of automotive electronic systems. In *Proceedings of 2nd Intl. ICSE workshop on Software Engineering for Automotive Systems (SEAS 05)*, 2005.

[9] Electronic Industries Alliance (EIA). Eia/is-731.1, systems engineering capability model, electronic industries alliance (interim standard), 2002.

[10] FlexRay Consortium. Flexray homepage, 2006. URL: http://-www.flexray.com/.

[11] International Standards Organisation (ISO). Iso 11898 road vehicles - interchange of digital information - controller area network (can) for high-speed communication, 1993.

[12] International Standards Organisation (ISO). Iso 11898-4. road vehicles - controller area network (can) - part 4: Time-triggered communication, 2000.

[13] R. Kazman, M. Klein, M. Barbacci, T. Longstaff, H. Lipson, and J. Carriere. The architecture tradeoff analysis method. In *Proceedings of the Fourth IEEE International Conference Engineering of Complex Computer Systems (ICECCS '98).* IEEE, 1998.

[14] G. Leen and D. Heffernan. Expanding automotive electronic systems. *IEEE Computer*, 35(1):88 – 93, January 2002.

[15] R. Nellore and R. Balachandra. Factors influencing success in integrated product development (ipd) projects. *IEEE Transactions on Engineering Management*, 48, May 2001.

[16] R. Nellore, K. Söderquist, and K. Å. Eriksson. A specification model for product development. *European Management Journal*, 17:50–63, 1999.

[17] O. Nierstrass, G. Arevalo, S. Ducasse, , R. Wuyts, A. Black, P. Müller, C. Zeidler, T. Genssler, and R. van den Born. A component model for field devices. In *Proceedings of the First International IFIP/ACM Working Conference on Component Deployment*, June 2002.

[18] Thomas Nolte, Lucia Lo Bello, and Hans Hansson. Facilitating subsystem integration by decoupling priority and identifier in can messages. In *Proceedings of the 5th International Workshop on Real-Time Networks (RTN'06) in conjunction with the 18th Euromicro International Conference on Real-Time Systems (ECRTS'06)*, July 2006.

[19] Thomas Nolte, Mikael Nolin, and Hans Hansson. Real-time server-based communication for can. pages 353–372. May 2006.

[20] OSEK/VDX. Osek/vdx operating system, 2006. URL: http://-www.osekvdx.org/mirror/os222.pdf.

[21] T. L. Saaty. How to make a decision. *The Analytic Hierarchy Process, Interfaces*, 24:19–43, November 1994.

[22] Siemens VDO. Siemens vdo website, 2006. http://usa.siemensvdo.com/-press/factsheet/ (Last Accessed: 2006-10-17).

[23] The Society of Automotive Engineers (SAE). Truck and bus control and communications subcommittee, sae j1939 standards collection, 2004.

[24] R. van Ommering, F. van der Linden, and J. Kramer. The koala component model for consumer electronics software. *IEEE Computer*, 33(3):78–85, March 2000.

Paper C

Making Decisions in Integration of Automotive Software and Electronics: A Method Based on ATAM and AHP

Peter Wallin
Mälardalen University
SE-72123 Västerås, Sweden
peter.wallin@mdh.se

Joakim Fröberg
Volvo CE
SE-63185 Eskilstuna, Sweden
joakim.froberg@volvo.com

Jakob Axelsson
Volvo Car Corporation
SE-40531 Gothenburg, Sweden
jakob.axelsson@mdh.se

Presented at SEAS, Minneapolis, USA, 2007

Abstract

In this paper we present a new method for making decisions on integration strategy for in-vehicle automotive systems. We describe the problem of choosing integration strategy and we describe the method, which is a combination of the Architecture Tradeoff Analysis Method, ATAM, and the Analytical Hierarchy Process, AHP. We exemplify the use of the proposed method by evaluating the integration decisions concerning the physical connection of a realistic example system; a computer controlled automatic gearbox. We present analysis on the use of the method and conclude that the method has several benefits compared to ATAM or AHP used individually. The method firstly supports a structured way of listing system goals, and secondly, it also supports the making of design decisions.

1 Introduction

Design of automotive in-vehicle electronic systems is a challenge for Original Equipment Manufacturers, OEMs, due to a large set of functional requirements and stringent quality goals. The system is required to deliver its many functions in a dependable and safe manner, and product costs are to be kept low. The system must fulfil business and life-cycle goals such as being simple to maintain, service, and produce. The resulting system architecture is often complex and system architecture design is a process with many stakeholders. One way of reasoning around architectural choices is to estimate quality attributes of the envisioned system and then try to quantify the impact of different choices.

1.1 Integration in automotive products

Design of automotive in-vehicle electronic systems includes joining together or integrating functionality developed by several organizations. These sub-systems can be purchased off-the-shelf from a supplier or developed specifically for its purpose by the OEM or the supplier, or a combination of the two. Functionality for sub-systems can be pure software like algorithms or it can be offered with hardware including computer nodes, sensors, actuators, connectors, etc. Integrating an electronic subsystem is the effort of making it conform to the decided architecture. Thus the integration is concerned with finding a design solution so that the component comply with, e.g. diagnostic strategy, system state management and fault handling. More precisely, integration could mean developing glue code or gateway functionality or it could mean to specify to a component supplier the system functionality to which the component must conform.

1.2 Problem description

OEMs often develop architectural guidelines based on the desired qualities and integration solutions should conform to these guidelines. Still integration is difficult. Either guidelines are too rigorous and need to be bent, or guidelines are too vague and fail to aid in design. Integration design, like architecture design, aims at finding a solution that meet many requirements from many stakeholders. This means that the system should not only be designed to provide its main function, but also to meet other requirements. For example, it is desired by the safety team that the system is feasible enough to analyze, and the service people wish for diagnostic functionality to cover all possible faults. Thus, the problem in integration is partly to know the various requirements and their importance, and partly to know what design is best suited.

1.3 Our proposed method

Our goal is to make the impact of integration decisions visible in terms of the desired properties of the system. Further we want to evaluate different integration strategies to find the one that best support the desired qualities of the product in its life cycle. In order to evaluate success of different integration strategies we need some criteria on how to decide what is favorable.

The approach of this work is to use scenarios from the Architecture Trade-off Analysis Method, ATAM [5], and analyze them with the Analytical Hierarchy Process ,AHP [10], to evaluate different integration strategies in the context of an automotive electronic system. Major research exists on both ATAM and AHP and both methods are quite commonly used [2, 3, 9].

The contribution of this work is the proposed method that combines ATAM and AHP, enabling structured reasoning and decision making. Although both methods are commonly used, still, there is to our knowledge no suggestion on how the two methods may be combined even if the possibility is mentioned by [11]. The method is applied to and intended for the context of automotive software and electronic systems, and more specifically we apply it to the decision making in choosing integration strategies. Although this paper focus on a limited number of integration strategies we believe that it can be used for all kind of integration strategies as well as other architectural decisions.

To demonstrate our approach we use an example concerning integration of a gearbox for construction equipment vehicles such as haulers, wheel loaders, and excavators. The example is simplified but has realistic specifications.

The rest of the paper is organized as follows. Section 2 introduces vehicle electronic systems. The properties of a vehicle electronic system is outlined in Section 2.1 and the four different integration strategies are presented in Section 2.2. We introduce a gearbox example in Section 2.3. Section 3 describes the proposed method. In Section 4 we provide a theoretical but realistic example of how the method will work. In Section 5 we analyze the method. Section 6 concludes the paper.

2 Vehicle electronic systems

In this section we present the context of automotive in-vehicle electronic systems. Further, we describe the notion of integration strategies and we provide a theoretic example of an automotive electronic system intended for integration based on previous studies.

2.1 General properties

Automotive electronic systems are safety critical, real time systems embedded in mechatronic components. The functions in an automotive vehicle include control of the engine and drive train, driver interface, suspension, comfort functions such as climate control, and audio/video systems. Besides the user functionality of the vehicle, there are numerous functions inside a vehicle that supports the production and service operations in the lifecycle of the product such as diagnostics and test. Sometimes the system and functionality is described as partitioned into subdomains, such as, powertrain, body, chassis, and infotainment. The implementation of the functionality in contemporary vehicles includes distributed computers with I/O to sensors and actuators. Wiring is substantial and bundled in cable harnesses. Control software is often constructed using a dataflow model and communication is often based on the CAN protocol.

In-vehicle computer systems are often labeled electronic systems in automotive applications. Automotive electronics thus includes electronic hardware such as sensors, actuators, Electronic Control Units (ECUs), and wiring, but also the software. The reason for using this term may be the close dependency of software and hardware in many automotive applications. For instance, a braking application is very tightly bound to the hardware for which it is tested and developed. A change of sensors or other hardware components in such an application would likely generate a change of software functionality. In the following we use the term electronic system to refer to the complete in-vehicle computer system including both software and hardware.

2.2 Integration strategies

Integration of new functionality is an iterative process. New functionality is added to an existing platform during many years. The same platform is also used for many different models and even different products.

Decisions on integration strategy will affect the quality outcome and lifecycle cost of not only the electronic system, but the complete vehicle. Integrating supplier electronics in automotive networks is challenging because several qualities are pursued simultaneously, much like in architecture design.

An integration strategy provides answers to questions on how a component will be made to fit into system wide schemes and principles. It is the design of interfaces and semantics of interaction between component and system. There may be several schemes to follow such as diagnostic signaling, fault handling, and state management. The component and its function can give rise to ways of interacting that are not covered by the decided system principles and schemes. An example is a mechatronic brake with many fault states that each affect the system state differently. Such issues are included in the integration strategy.

Network topology decisions is part of the integration strategy. To describe the method of evaluating integration strategies we focus on how a function is to interface the system. The four alternatives we consider in this paper are shown in Figure 1 and are explained in the list below.

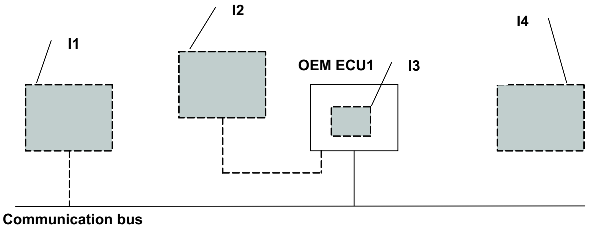

Figure 1: **Four choices in integration strategy**

I1. New ECU connected directly on a system bus.
I2. New ECU connected via a gateway.
I3. Application software component located in existing ECU.
I4. New ECU stand alone - not connected to a bus.

2.3 Example: Gearbox

Thus, new ECUs contains both a new software functionality and a software environment including operating system, device drivers, and possibly more. Integration strategy I3 on the other hand involves only the software function without surrounding infrastructure. Based on a previous study of three cases of real-life mechatronic integration [4], we have developed a theoretical but realistic example of a component intended for integration in an automotive application. The example consists of a mechanical gearbox with a fitted ECU that controls the operation of the automatic gear shifting intended for use in a construction equipment vehicle.

The ECU is equipped with the following interfaces:

- A CAN interface
- J1939 [1]
- A serial interface with a proprietary protocol for diagnostics

The gearbox application is dependent on signals that describe the gear lever position, engine speed, vehicle speed, and drive mode. The application must be able to control engine speed for short periods of time during gearshifting. There are timing requirements on the control messages; latency, periodicity, and jitter are specified. The application also has a number of error states where gearshifting is not possible.

3 The method explained

ATAM is a method for identifying important design decisions and show how they tradeoff against each other in software architectures. AHP is a multi criteria analysis method. By combining the two methods we can use scenarios produced by ATAM as input to AHP and carry out a robust evaluation of both scenarios and how well an integration strategy fits a certain scenario. In this section, we briefly summarize the original methods and theen comment on how we combine them for decision support in an automotive Electrical/Electronic architecture.

3.1 ATAM

The goal of ATAM is to assess the consequences of architectural decisions in the light of quality attribute requirements [5]. Typically there exist competing quality attributes such as modifiability, security, reliability and maintainability that different stakeholders consider to be the most important. These quality attributes are broken down into scenarios. ATAM is divided into nine steps. These steps involve eliciting a utility tree and identifying risks, sensitivity and tradeoff points.

In our approach we only consider some of the steps in ATAM and it is mostly how the scenarios in the utility tree are generated that is of relevance in the proposed method. The complete description on ATAM can be found in [5].

3.2 AHP

The Analytic Hierarchy Process (AHP) is a multi-criteria decision making approach in which factors are arranged in a hierarchic structure [10]. In AHP all element are compared against each other which yield a robust result but also time consuming due to the large number of comparisons. Elements are compared according to Table 1. In this paper we use an AHP related approach called Chainwise Paired Comparison (CPC) [8]. CPC only requires the same amount of comparisons as the number of elements. However the consistency needs to be validated to ensure the same result as with AHP. The CPC algorithm is shown in Table 2 which is adapted from Table 1 in [8].

Table 1: Element comparison

Scale	**Importance**
1	Equal importance
3	Moderate importance of one over another
5	Essential or strong importance
7	Very strong importance
9	Extreme importance
2,4,6	Intermediate values

We are interested in for n elements finding the weight W_i. Since it is difficult to estimate this weight directly, we instead ask the decision maker for the ratio R_i between two successive elements as shown in Equation 1.

$$R_i = \begin{cases} \dfrac{W_i}{W_{i+1}} & : i = 1..n-1 \\ \dfrac{W_n}{W_1} & : i = n \end{cases} \tag{1}$$

D_i represents the estimated value of the ratio R_i. If the estimate is perfect then Equation 2 is true, meaning that the estimates are consistent.

$$\prod_{j=1}^{n} D_j = 1 \tag{2}$$

Full consistency can be hard to achieve in practice with many factors to chainwise compare. To compensate for this inconsistency we compute a new estimated ratio,

Table 2: **Algorithms used in chainwise paired comparison**

i	R_i	D_i	$\tilde{R}_i$	M_i	V_i
1	$\frac{W_1}{W_2}$	D_1	$\frac{D_1}{\sqrt[n]{\prod D_j}}$	$\tilde{R}_1 \cdot M_2$	$\frac{M_1}{\sum M_j}$
2	$\frac{W_2}{W_3}$	D_2	$\frac{D_2}{\sqrt[n]{\prod D_j}}$	$\tilde{R}_2 \cdot M_3$	$\frac{M_2}{\sum M_j}$
:	:	:	:	:	:
n-1	$\frac{W_{n-1}}{W_n}$	D_{n-1}	$\frac{D_{n-1}}{\sqrt[n]{\prod D_j}}$	$\tilde{R}_{n-1} \cdot M_n$	$\frac{M_{n-1}}{\sum M_j}$
n	$\frac{W_n}{W_1}$	D_n	$\frac{D_n}{\sqrt[n]{\prod D_j}}$	1	$\frac{M_n}{\sum M_j}$

$\tilde{R}_i$, with Equation 3. $\tilde{R}_i$ is by definition a consistent estimation, fulfilling Equation 2.

$$\tilde{R}_i = \frac{D_i}{\sqrt[n]{\prod_{j=1}^{n} D_j}} \tag{3}$$

Assume that M_i represent W_i/W_n and since $\tilde{R}_i$ is an estimate of R_i, M_i can now be computed recursively with equation 4.

$$\begin{cases} M_i = \tilde{R}_i \cdot M_{i+1} \\ M_n = 1 \end{cases} \tag{4}$$

We now have a weighted list of elements. To make values comparable to each other we normalize the weights with Equation 5.

$$V_i = \frac{M_i}{\sum_{j=1}^{n} M_j} \tag{5}$$

3.3 The proposed method

We have devised a method, based on a combination of ATAM and AHP, that allow us to find the best choice out of a number of possible designs. The basic steps in the method are shown below, and later exemplified with more details in the next section.

1. Elicit scenarios from system stakeholders
2. Rate importance of scenarios
3. Assess scenario fulfilment of each design choice

Elicit scenarios from system stakeholders. Using some of the basic steps of ATAM, a list of scenarios is extracted. Each scenario represents an important aspect of the system that is desired in order to achieve a "good" system. What constitutes a good system depends on who you ask, and therefore, the ATAM stipulates to involve many stakeholders that has interests in the systems life cycle as well as experienced system architects. The scenarios that come from this elicitation can be grouped in a tree structure called a utility tree, and in this way the scenarios can be shown to belong to a certain quality attribute such as reliability. This work involves interviews and workshops and can be substantial. However, the resulting set of scenarios is a general characterization of the system requirements in terms of qualities. Thus, it is not only usable for a particular decision. As the life cycle of an automotive product is different for different companies, it seems unrealistic to elicit a general utility tree even for a certain kind of vehicle. The generality of the scenarios is likely confined to the company and possibly to the type of vehicle, e.g., a minivan or sports car. The ATAM stipulates a procedure for prioritizing scenarios and this can be used to shorten the possibly long list of scenarios.

Rate importance of scenarios. A more formal prioritization and weighting of scenarios can be made by employing the AHP procedure. Comparing each scenario to all others to get a weighting is possible and the most accurate method for AHP prioritization. Since the number of comparisons required with AHP are $n(n-1)/2$ we get, even with a small set of scenarios an extensive list of comparisons. We instead propose to use chainwise paired comparison as shown in [8], to reduce the number of comparisons to n. Chainwise comparison is made by comparing the first scenario with the following in the list. This is continued for all scenarios and finally the last scenario is compared to the first to get a "chain". Each comparison is made using the AHP method scores that are shown in Table 1. This procedure yields a weight for each scenario that corresponds to the importance of that scenario.

Assess scenario fulfilment of each design choice. Here, we have to have a number of defined design choices. For each design choice, a fulfilment is estimated of each scenario i.e. it should be estimated how well each design choice meets each scenario. For instance a simple design may score high on a scenario requesting ease of safety analysis. More in detail, each design decision is compared to another in chainwise manner until all have been visited and the last compared to the first. What this gives us, after AHP prescribed calculations, is a weight for each design decision. The weight corresponds to how well that design meets the selected scenario. So, for a set of four defined design alternatives and 16 scenarios, we get a sum of $4 * 16$ weights. The final step in finding the best solution is then calculated by using the weight (importance) of each scenario. Now, we know the "goodness" of the design choice with respect to each scenario, and we also know the importance of each scenario. We add up the product of scenario weight and design choice weight for all scenarios. This number corresponds to how much fulfilment of all the scenarios that this particular design decision has, and thus we have comparable numbers for the set of design decisions. This final step is not general, but the estimations of fulfilment must be made for a certain automotive product, for a certain component to be integrated.

4 Using the method

In this section we explain how the method can be used. The gearbox from the example in Section 2.3 is to be integrated with one of the four different integration strategies explained in Section 2.2.

The ATAM proposes that this elicitation is done in two workshops including all key personnel. For practical reasons, we have deviated from the stipulated workshop format and elicited a utility tree based on four interview sessions with only two experts individually. First we use interview results from previous work on quality attributes in automotive electronics and software systems [6][7]. We use these results to construct an initial utility tree which is then used to guide another round of interviews. This round yields a set of scenarios that we use in our following theoretical example.

4.1 Scenarios

ATAM states that "A scenario is a short statement describing an interaction of one of the stakeholders with the system". Here we list the scenarios that we elicited from the interviews with architects and product specialists. The respondents described the business situation related to each quality attribute. This list is not at all a complete list of scenarios that should be considered but for explaining the method we find it sufficient. In order to extract a complete list, we would like to include all stakeholders and also fully utilize the workshop format proposed in ATAM.

Below is the list of scenarios that where elicited from the interviews categorized under their main utility.

Safety

S1. A safety related function experiences a fault and this does not lead to an unsafe state of the system

S2. The system experiences a fault and each safety related function can reduce functionality according to a system wide policy

S3. Each safety related function does not add any non recoverable unsafe states (e.g. loss of steering is difficult to recover safely from)

S4. Safety analysis is performed and the logics of each safety related function is visible for inspection

Reliability

S5. Overall reliability benefits from certified or tested physical criteria - EMC, moist, dust, vibration and shock

S6. A fault occurs and fault tolerant design upholds system function

S7. Minimum number of connectors wanted

S8. Testable design wanted

S9. Simpleness preferred

S10. Fault diagnosis desired

Modifiability

S11. A function is to be reused in a new vehicle project and the system functionality partitioning is different

S12. A function is to be reused in a new vehicle project and different networks and protocols are to be used

S13. Porting SW platform to new hardware

Serviceability

S14. A function is faulty and the on-board diagnostic system finds the root case of the problem (e.g. eroded connector or faulty sensor)

S15. Physical components are easily replaced

S16. Software functionality is easily replaced

Table 3: **Scenarios prioritized with chainwise paired comparison**

i	R_i	D_i	I_i	$\tilde{R}_i$	M_i	V_i
1	S_1/S_2	2	2,915	2,048	3,907	0,090
2	S_2/S_3	$\frac{1}{5}$	0,292	0,205	1,908	0,044
3	S_3/S_4	1	1,458	1,024	9,318	0,213
4	S_4/S_5	7	10,204	7,167	9,101	0,208
5	S_5/S_6	$\frac{2}{5}$	0,583	0,410	1,270	0,029
6	S_6/S_7	7	10,204	7,167	3,101	0,071
7	S_7/S_8	$\frac{1}{3}$	0,486	0,341	0,433	0,010
8	S_8/S_9	$\frac{1}{3}$	0,486	0,341	1,268	0,029
9	S_9/S_{10}	1	1,458	1,024	3,715	0,085
10	S_{10}/S_{11}	2	2,915	2,048	3,628	0,083
11	S_{11}/S_{12}	3	4,373	3,072	1,772	0,041
12	S_{12}/S_{13}	1	1,458	1,024	0,577	0,013
13	S_{13}/S_{14}	$\frac{2}{7}$	0,437	0,307	0,563	0,013
14	S_{14}/S_{15}	7	10,204	7,167	1,834	0,042
15	S_{15}/S_{16}	$\frac{1}{4}$	0,364	0,256	0,256	0,006
16	S_{16}/S_1	$\frac{1}{4}$	0,125	0,239	1,000	0,023

4.2 Prioritizing the scenarios with chainwise paired comparison

Here the 16 scenarios are prioritized with CPC. In this example we assume that the 16 scenarios elicited from the interviews are the most important ones. Asking the full set of stakeholders the number of scenarios could have been significantly larger. The lowest prioritized scenarios would then be discarded as not important enough to affect the choice of integration strategy. In Table 3 the scenarios are chainwised compared. It is only the value D_i that is manually estimated according to Table 1 in Section 3.2. All other values are calculated with the equations in Table 2. V_i is the calculated priority. In this theoretical gearbox example S_3 is considered to be the most important scenario and will therefore have higher impact when integration strategy is chosen.

As explained in Section 3.2 we need to check if the system is consistent. In this example the consistency is calculated to 98%. Table 3 in [8] shows that for 16 elements the consistency needs to be at least 95.7% for the data to be valid.

Table 4: **Scenario S8**

i	R_i	D_i	I_i	$\tilde{R}_i$	M_i	V_i
1	I_1/I_2	2	2,400	2,093	2,866	0,274
2	I_2/I_3	$\frac{1}{4}$	0,300	0,262	1,369	0,131
3	I_3/I_4	5	6,000	5,233	5,233	0,500
4	I_4/I_1	$\frac{1}{3}$	0,400	0,349	1,000	0,096

4.3 Weighting scenarios against an integration strategy

Each scenario is now weighted against the four different integration strategies. After this comparison we have a prioritized list of all scenarios and also one list per scenario showing how well each integration strategy meets the particular scenario. Displayed in Table 4 is how well scenario S_8 correspond to each of the four integration strategies. The final analysis is done by using the weight V_i of each scenario and multiply it with the weight of how well it is supported by each integration strategy. This is shown in Table 5. The integration that seems to be most suitable in the gearbox example is integration strategy I_3.

5 Analysis

The goal of this work is to find a feasible method that can be used in practical cases of decision making in the context of integration of automotive electronics.

5.1 The method compared to AHP and ATAM

The method does provide a structured way of using expert knowledge to make decisions in design of automotive electronics and possibly many other areas. Like ATAM recognizes, the difficulties in making decisions stems from the complexity where many stakeholders have different goals. What ATAM lacks is the actual support for decision making. ATAM is instead intended to identify sensitive design points in the system, but choosing a design alternative must be done by other means. AHP on the other hand is a method for decision making with multiple criteria, but lacks a structured way of listing the criteria. Thus, using the concept of scenarios and utility trees from ATAM as input to an AHP process gives us a method that includes both benefits. Compared to using ATAM alone, the combined method supports decision making and should still have the benefits that

Table 5: **Decision matrix**

		I_1	I_2	I_3	I_4
S_1	0,090	0,077	0,154	0,077	0,692
S_2	0,044	0,321	0,321	0,321	0,036
S_3	0,213	0,370	0,185	0,370	0,074
S_4	0,208	0,067	0,081	0,686	0,166
S_5	0,029	0,125	0,125	0,625	0,125
S_6	0,071	0,286	0,143	0,429	0,143
S_7	0,010	0,227	0,160	0,453	0,160
S_8	0,029	0,274	0,131	0,500	0,096
S_9	0,085	0,273	0,154	0,086	0,486
S_{10}	0,083	0,364	0,182	0,364	0,091
S_{11}	0,041	0,125	0,125	0,625	0,125
S_{12}	0,013	0,127	0,301	0,537	0,035
S_{13}	0,013	0,113	0,126	0,556	0,205
S_{14}	0,042	0,286	0,143	0,286	0,286
S_{15}	0,006	0,222	0,222	0,111	0,444
S_{16}	0,023	0,174	0,162	0,602	0,062
Final priority		**0,227**	**0,153**	**0,414**	**0,205**

has been reported with ATAM. One such important benefit is that stakeholders get to reason about qualities and their fulfilment. Thus, compared to using AHP alone, we will get both a structure for the criteria and likely also the benefit of stakeholder involvement and communication.

5.2 Methods pros and cons

One of the main problems with multi criteria decisions is to find out the relative importance of each goal. To investigate this, a number of estimates must be made by experts. It is much desired to keep the number of estimations low to get a feasible method. The AHP method prescribes comparing and estimating the relative importance of each criteria against all other, and thus having a matrix of estimations to perform with $n(n-1)/2$ estimations. For weighting the importance of the scenarios, we chose to perform chainwise paired comparison that reduces the number of comparisons to n. It should be noted though that the weighting of scenarios is something that can be reused for other decisions. A large effort in weighting scenarios could be accepted if there are many decisions to make.

- **Flexible and scalable**. As we progress through the method we can choose to employ more or less rigorous comparisons depending on the importance of the design decision. For instance it may be justified to employ the full comparison scheme as opposed to the chainwise, if we would want to integrate a new engine system with high impact on system behaviour. Likewise we can choose to have a high number of scenarios if the decision is judged very important.

- **Feedback on accuracy**. The AHP calculations produce a measure of consistency for the estimations made by the experts. Thus, both in the second and third step we will get feedback on whether the interviews have been successful. If the consistency is too low, we can instead decide to redo some of the importance assessments.

- **The method has some support for answering why**. An important issue when designing systems is to have an understanding by all involved why a certain design has been chosen. If the "why" is clearly understood, we run a low risk that the decision is overrun by a new decision. It is clearly visible in the AHP process how the relative importance measures have been estimated. This would likely aid in the effort of explaining why decisions have been made.

6 Conclusions

In this paper, we have presented a new method for making decisions on integration strategy for in-vehicle automotive systems. The method is based on a combination of the Architecture Tradeoff Analysis Method, ATAM, and the Analytical Hierarchy Process, AHP. We have described the method in detail and exemplified its use with a theoretical but realistic example of an electronic controlled gearbox that is to be integrated into an in-vehicle electronic system. Analyzing the method and the example, we have shown that the method is usable and has benefits compared to either ATAM or AHP used individually. Like ATAM, this method provides a way for stakeholders to reason about system qualities, but it does not stop at identifying important design points. Compared to using ATAM alone, our combined method supports decision making and should still have the benefits that has been reported with ATAM. One such important benefit is that stakeholders get to reason about qualities and their fulfilment. Thus, compared to using AHP alone, we will get both a structure for the criteria and likely also the benefit of stakeholder involvement and communication.

In analyzing the method and the example, we have also shown that the method seems feasible and that it supports some desired properties. Firstly, it is scalable in effort to compensate for more or less crucial decisions. Secondly, we show that it provides feedback on the quality of the estimates. Thirdly, the method does provide some documentation as to why a decision has been made and this possibly helps in understanding and communicating system design among stakeholders.

References

[1] SAE J1939 Standard - The Society of Automotive Engineers (SAE) Truck and Bus Control and Communications Subcommittee, SAE J1939 Standards Collection, 2004 Available at: http://www.sae.org/products/j1939.htm.

[2] M. Barbacci, P. Clements, A. Lattanze, L. Northrop, and W. Wood. Using the architecture tradeoff analysis method (ATAM) to evaluate the software architecture for a product line of avionics systems: A case study. Technical note, CMU/SEI-2003-TN-012, 2003.

[3] J. K. Bergey and M. J. Fisher. Use of the architecture tradeoff analysis method (ATAM) in the acquisition of software-intensive systems. Tecnical note, CMU/SEI-2001-TN-009, 2001.

[4] J. Fröberg and M. Åkerholm. Integration of electronic components in heavy vehicles: A study of integration in three cases. In *Proceedings from Systems Engi-*

neering/Test and Evaluation Conference, Melbourne, 25-27 September 2006, Melbourne, September 2006.

[5] R. Kazman, M. Klein, and P. Clements. ATAM: Method for architecutre evaluation. Technical report, Carnegie Mellon Software Engineering Institute, CMU/SEI-2000-TR-004.

[6] A. Möller, M. Åkerholm, J. Fröberg, and M. Nolin. Industrial grading of quality requirements for automotive software component technologies. In *Embedded Real-Time Systems Implementation Workshop in conjunction with the 26th IEEE International Real-Time Systems Symposium*, 2005 Miami, USA, December 2005.

[7] A. Möller, J. Fröberg, and M. Nolin. Industrial requirements on component technologies for embedded systems. In *International Symposium on Component-based Software Engineering (CBSE7)*, Edinburgh, Scotland, May 2004. Springer Verlag.

[8] J. W. Ra. Chainwise paired comparison. *Decision Sciences*, 30(2):581–599, 1999.

[9] G. C. Roper-Lowe and J. A. Sharp. The analytic hierarchy process and its application to an information technology decision. *The Journal of the Operational Research Society*, 41(1):49–59, jan 1990.

[10] T. Saaty. *The analytic hierarchy process*. New York: McGraw-Hill, 1980.

[11] L. Zhu, A. Aurum, I. Gorton, and R. Jeffery. Tradeoff and sensitivity analysis in software architecture evaluation using analytic hierarchy process. *Software Quality Control*, 13(4):357–375, 2005.