Journal Name etc. manuscript No.
(will be inserted by the editor)

Joakim Fröberg · Mikael Åkerholm · Kristian Sandström · Christer Norström

Key Factors for Achieving Project Success in Integration of Automotive Mechatronics

Received: date / Accepted: date

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.

Keywords Integration · System architecture · Automotive · Embedded systems

J. Fröberg
Volvo Construction Equipment
Eskilstuna, Sweden
E-mail: joakim.froberg@volvo.com

M. Åkerholm, K. Sandström, C. Norström
Mälardalen University
Västeras, Sweden
E-mail: mikael.akerholm@mdh.se

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*" [23] is stated by one of the worlds 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

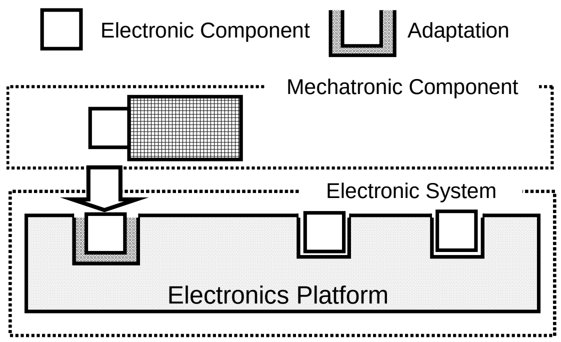

Fig. 1 In-vehicle electronic system design by integration

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 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 [24] 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 [21] 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

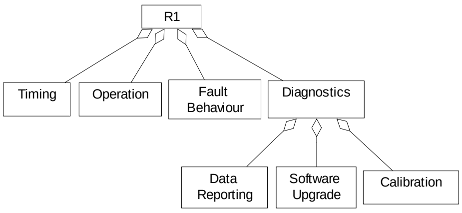

Fig. 2 Checkpoints for Recommendation 1

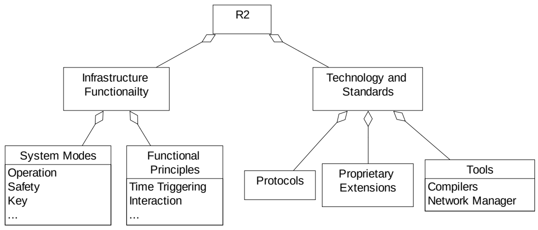

Fig. 3 Checkpoints for Recommendation 2

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 [20] 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 sub-systems 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.

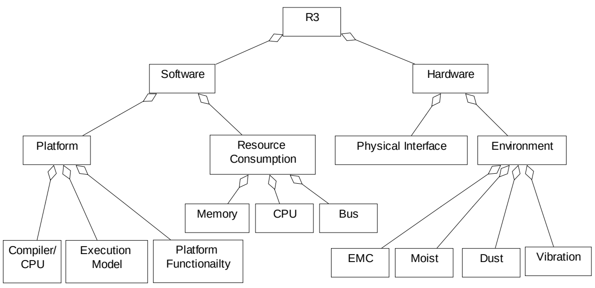

Fig. 4 Checkpoints for Recommendation 3

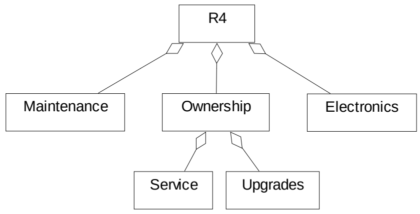

Fig. 5 Checkpoints for Recommendation 4

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 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

Table 1 Early electronics functionality decisions

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

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. 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.

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

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

Table 3 Platform requirements decisions

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

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 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.

Table 4 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

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 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.

Table 5 Integration solution

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

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.

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

5.1 Case data

In fig. 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 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

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**

Fig. 6 Case characteristics

Table 7 Early stakeholder involvement

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

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.

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

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.

Table 9 Project Success

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

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.

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 4.

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

Table 10 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 11 Filfilment of recommendations

	Overall fulfilment R1	Overall fulfilment R2	Overall fulfilment R3	Overall fulfilment R4	Total fulfillment average	Total measure of 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

late involvement of the electronics department is a problem and it seems logical to assume that this is an important stakeholder. 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 10, and put into Table 9.

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.

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 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

Table 12 Project characteristics

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

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. 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 picure 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) [22]. 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. Åkerholm M, Carlson J, Fredriksson J, Hansson H, Håkansson J, Möller A, Pettersson P, Tivoli M (2006) The save approach to component-based development of vehicular systems. Journal of Systems and Software Available online 26 September 2006, Elsevier
2. Åkerholm M, Carlson J, Fredriksson J, Hansson H, Nolin M, Nolte T, Håkansson J, Pettersson P (2006) 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)
3. Auer G (2004) Automotive news europe, mercedes will make cars less complicated, technology leader cuts 600 electronic functions, seeks more. May 31, http://-www.accessmylibrary.com/coms2/summary_0286--21549689_ITM (Last Accessed: 2006-10-17)
4. AUTOSAR (2006) Homepage of automotive open system architecture (autosar). URL: https://www.decos.at/ (Last Accessed: 2006-01-17)
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. Casparsson L, Rajnak A, Tindell K, Malmberg P (1998) Volcano - a revolution in on-board communication. Tech. rep., Volvo Technology Report
7. DECOS (2006) Dependable embedded components and systems. URL: https://www.decos.at/ (Last Accessed: 2006-01-17)
8. Eklund U, Askerdahl, Granholm, Axelsson J (2005) 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)
9. Electronic Industries Alliance (EIA) (2002) Eia/is-731.1, systems engineering capability model, electronic industries alliance (interim standard)
10. FlexRay Consortium (2006) Flexray homepage. URL: http://www.flexray.com/
11. International Standards Organisation (ISO) (1993) Iso 11898 road vehicles - interchange of digital information -

controller area network (can) for high-speed communication
12. International Standards Organisation (ISO) (2000) Iso 11898-4. road vehicles - controller area network (can) - part 4: Time-triggered communication
13. 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
14. Leen G, Heffernan D (2002) Expanding automotive electronic systems. IEEE Computer 35(1):88 – 93
15. Nellore R, Balachandra R (2001) Factors influencing success in integrated product development (ipd) projects. IEEE Transactions on Engineering Management 48
16. Nellore R, Söderquist K, Eriksson KÅ (1999) A specification model for product development. European Management Journal 17:50–63
17. Nierstrass O, Arevalo G, Ducasse S, , Wuyts R, Black A, Mller P, Zeidler C, Genssler T, van den Born R (2002) A component model for field devices. In: Proceedings of the First International IFIP/ACM Working Conference on Component Deployment
18. Nolte T, Bello LL, Hansson H (2006) 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)
19. Nolte T, Nolin M, Hansson H (2006) Real-time server-based communication for can. pp 353–372
20. van Ommering R, van der Linden F, Kramer J (2000) The koala component model for consumer electronics software. IEEE Computer 33(3):78–85
21. OSEK/VDX (2006) Osek/vdx operating system. URL: http://www.osekvdx.org/mirror/os222.pdf
22. Saaty TL (1994) How to make a decision. The Analytic Hierarchy Process, Interfaces 24:19–43
23. Siemens VDO (2006) Siemens vdo website. Http://-usa.siemensvdo.com/press/factsheet/ (Last Accessed: 2006-10-17)
24. The Society of Automotive Engineers (SAE) (2004) Truck and bus control and communications subcommittee, sae j1939 standards collection