Integration of Electronic Components in Heavy Vehicles: A Study of Integration in Three Cases

Joakim Fröberg
Volvo Construction Equipment
Components Division
SE 631 85, Eskilstuna, Sweden
joakim.froberg@volvo.com

Mikael Åkerholm
Mälardalen University
MRTC, IDE
SE 721 23, Västerås, Sweden
mikael.akerholm@mdh.se

Abstract. Complexity of in-vehicle computer systems and the availability of computerized mechatronics yield a situation where automotive electronic systems are designed by integration. Since the qualities of a modern vehicle are much dependent on the in-vehicle computer system, integration is a major issue which has proved difficult with respect to assessing quality and cost. OEMs of automotive products want leverage over targeted qualities and the cost of scale when purchasing supplier components.

In this paper, we present three cases of integration of mechatronic components into vehicle platforms of Volvo Construction Equipment and focus on the integration of embedded computer systems. The study shows problems areas of communicating architecture constraints, evaluation of components in early phases of development, and lacking definitions of responsibilities.

Based on the study, we list four recommended practices to avoid the problems found in the cases. The analysis shows that integration of embedded computers should be considered early in development and in order to reduce project risks, the early assessment of computers need be fairly detailed. From the study we also present driving requirements in design of in-vehicle computer systems.

Keywords. Integration, System architecture, Automotive, Embedded systems

INTRODUCTION

In this paper we present our findings from an industrial case study on integration of embedded computer components in heavy vehicles. The study consists of three cases where a mechatronic component has been integrated into a construction equipment machine as part of a development project. The embedded computer part of the mechatronic component has been integrated into existing electronic vehicle platforms; all cases are from the same company Volvo Construction Equipment, VCE.

The effort of integration puts focus on system level functionality and architecture of the system. Safety, reliability and maintainability are qualities that are normally rated very important by automotive OEMs and integration stresses the way we design to meet such targeted qualities.

This study is part of a continuous improvement program within the company and the purpose of the study is to explore how embedded computers and software components are integrated into an existing system, and what efforts are made to achieve targeted qualities. From this work we identify problem areas for further study.

The contribution of this work is the list of recommended practices supported by findings from the study. The description of industrial cases and the driving requirements in automotive in-vehicle computer system design. Moreover, our contribution is the open issues and directions for future studies.

The rest of the introductory parts include section 1.1 with background on automotive system integration, and section 1.2 with a problem description. In Section 2 we present the three cases. Section 3 contains our findings from the study with findings and recommended practices. Section 4 discusses related works, in terms of possible system integration approaches. Section 5 discuss future work, and eventually Section 6 contains our conclusions.

Background. 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 sources. Components from external sources 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 computers. Examples are brake-, engine-, hydraulic-, and climate-systems, all which typically include advanced computer systems. Also, typically these computer 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 computer 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 system architecture e.g., 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 OEM in-vehicle computer platform is a set of design decisions, components, process and tools that is reused between vehicles [1].

The architectural choices made by the OEM are manifested in the platform. Examples are operating systems, communication buses, 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 system with desired qualities and cost targets and makes different architectural choices. Because of the possible architectural mismatch when joining the two, the electronic component can conform more or less well to its intended environment.

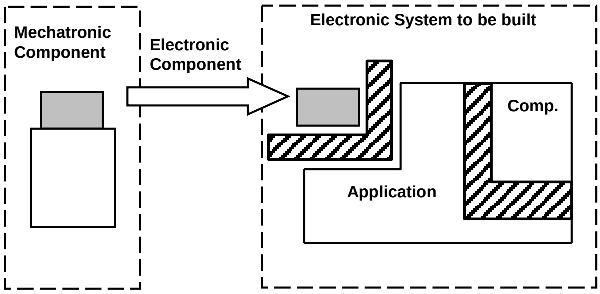

Figure 1. In-vehicle electronic system design by integration.

So, 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. In this paper we refer to the process of doing this design as integration.

Given an off-the-shelf component and an architecture there may not exist a feasible integration solution. The phase of integration also includes redesign to remove constraints. 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 dashed area in Figure 1.

Problem and Objectives. System integration has proved difficult with respect to assessing quality and cost. OEMs want leverage over qualities and the cost of scale of purchasing supplier components. The problem is accentuated by the growth of complexity in automotive embedded systems [2].

An OEM used to develop computers and software in-house need to shift to a model of development more focused on system integration. First, the electronic system architecture needs to support integration. Secondly, technical and architectural solutions for integration need to be investigated with respect to qualities. Thirdly, the engineering methods of integration need to be decided.

Ultimately, OEMs would want to have predictions on which architecture, integration solutions, and methods that affect qualities. In this study we will start by looking into computer system architecture, integration and method in the three cases and further we will collect current practices. Moreover we identify problem areas.

The objectives of this study:

1. Study architectures, and integration solutions in industrial cases.
2. Identify problem areas suggest solutions.
3. Identify related work and collect current practices.
4. Suggest further directions of study.

THREE CASES OF INTEGRATION

We have studied three cases of mechatronic integration in Volvo Construction Equipment, VCE, product development.

Method. We performed interviews with senior technical staff involved in the three cases of integration at Volvo Construction Equipment. 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. The results were put in a table and compared and analyzed.

Some non-public documentation was provided during the interviews. This information however is not used in the reasoning we provide in the analysis.

The work has resulted in a Volvo internal report, a technical report, and this paper. The Volvo report is not public, but more details of the case study are available in the technical report [3]. In this paper the cases have been anonymized to not reveal which products correspond to the cases.

Volvo CE System architecture. The architecture of the electronic system is an important input when deciding on how a supplier component can be integrated as we have shown. Here we outline the VCE electronic platform and its architecture. For a more thorough description of the electronic architecture of Volvo Construction Equipment, see [4].

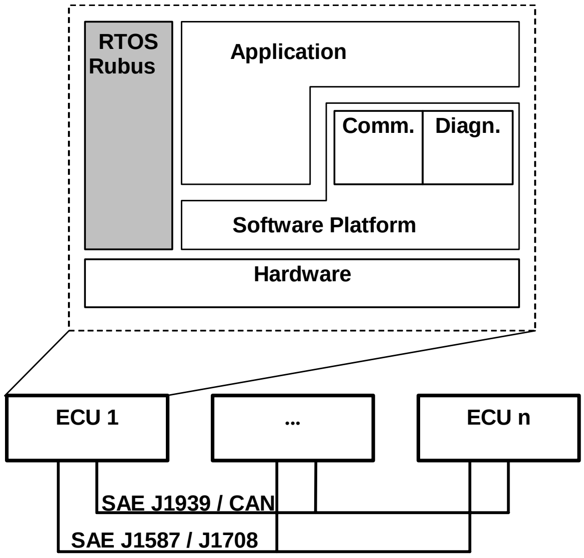

Figure 2. VCE Electronic platform.

The VCE system architecture uses the communication protocol SAE J1939 on CAN for control applications. The architecture allows subnets and redundant busses to be used. There is a slower bus based on SAE J1587 used for diagnostic signaling in the system.

Together with a hardware platform, VCE uses a common software platform for the on-board Electronic Control Units, ECUs. The platform includes communication and diagnostic services as well as infrastructure such as I/O handling and operating system.

Two basic constraints on integration solutions are; firstly there can be no messages that are unknown to VCE on control busses and, secondly that each ECU bandwidth must be bounded and known.

A word on quality goals. During interviews, the respondents have discussed quality at length. Here, we summarize the key qualities desired by automotive OEMs. In our study respondents evaluate outcome in terms of quality largely with these three qualities; safety, reliability, and maintainability.

Safety – “Safety is freedom from accidents or losses” [5]. In computer system terms, safety is to keep the system in a safe state always. To do this a system should employ mechanisms to be informed on failures and make sure functional degradation is done without risk. To an automotive OEM this has several implications; failure detection mechanisms, safe design as well as a design that facilitates safety analysis are desired.

Reliability – The IEEE defines reliability as "The ability of a system or component to perform its required functions under stated conditions for a specified period of time" [6]. The ability to provide function over a time often means to provide its function without failures, which means that faults should not propagate to loss of function. Thus, design for reliability includes, apart from high quality components, redundancy and fault handling schemes. Providing a reliable product is a high priority to OEMs and unplanned services are undesired.

Maintainability – The IEEE defines maintainability as “The ease with which a software system … can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment” [6] In the context of automotive computer systems, a maintainable solution includes both the ability to perform further development and to be able to service the vehicle during its lifecycle. Further development involves new products and technology change e.g. changing ECU hardware. Servicing a vehicle includes maintenance functionality like fault diagnostics and software upgrades. For example a node that doesn’t provide diagnostics of faulty hardware parts and does not allow software upgrades can be costly to service.

CASES

We have studied the integration of three mechatronic components in three different projects at Volvo Construction Equipment. Here we provide a description of the electronics integration of each: project context, integration, and problems reported.

Case 1: Software component. This project introduced computer controlled mechanics related to the drive train. A supplier offers a system with mechanical components as well as control system including sensors, actuator, computer hardware and software. VCE chose 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 VCE software platform.

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 VCE. Therefore the supplier compiled the code and delivered a binary that was verified at VCE. This process was repeated several times and neither the supplier nor VCE could debug the component in its real environment.

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. The only real drawback to maintenance is the effort of testing that must be done when a change in software platform or software component is done.

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.

Problems encountered during project: There were unforeseen difficulties in verification to deal with and as a result a late project. The quality of the specification was inadequate when verification began and had to be improved during the course of the project.

Case 2: Climate control ECU. This project developed a modular solution to provide a climate control in the cabins of machines. 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 VCE and contains a software platform with operating system and communication software components. The ECU has two CAN interfaces and one 1587 interface.

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.

For the computer system, specifications were not adequate and needed improvement in order to get through the process of verification. However, 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 VCE standard 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.

Problems encountered during project: Very minor. The project was on time and on cost.

Case 3: Hydraulic control ECU. The objective of this project was to integrate a computer controlled hydraulic component to achieve a hydraulic function. The embedded computer system consisted of an ECU with a control application and one CAN interface. Also included was an angle 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.

Achieving a technical integration solution for this system presented several difficulties. The sensor and the ECU both required connection via CAN. Also, the system required system data sent on the VCE control bus. Further, the communication between sensor and ECU required substantial bandwidth. Another constraint was that the ECU implemented a diagnostic protocol that was not compatible with VCE requirements. These constraints together show a case where the impact on the electronic platform is high. The solution was not finally decided at the time of this study but the problems had delayed the project substantially.

Problems encountered during project: Many issues turned up and had to be handled during 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.

Comparison of parameters from cases. We summarize the findings on differences in Table 1

Case	Initial quality of specs	Impact on the electronic platform	Safety criticality	Impact on overall Product behavior
1	Low	Medium	Medium	High
2	Low-Medium	Low	None	Low
3	Low	High	High	High

Table 1: Comparison of cases

Specifications are shown to be inadequate in all three cases. As indicated by the table the interviews reveal that the levels of integration in the three cases are different. Case 1 and 3 involves components that have a relatively high impact on the electronic system as well as having bearing on safety and product behavior. Case 2 on the other hand has lower impact in terms of electronic platform, product behavior and safety.

We see that case 2 which was successful has a different profile compared to case 1 and 3. The integration of case 2 has an overall lower impact on the product. This fact has likely affected the outcome in the positive direction. The quality of specifications in case 2 is also somewhat higher, but still reported inadequate by respondents.

LEARNINGS AND ANALYSIS

In summary, we have two cases that had deviations from plan and one that was largely successful. Case 2 is on time and budget, while Case 1 and 3 is delayed and did meet unforeseen problems in order to achieve the targeted functionality and quality.

Findings from interviews. An overview comparison of the cases is provided here by presenting the findings of each topic of the interviews.

General - In asking general questions on which organizations were involved in the cases it became obvious that many departments within the OEM makes decisions that impacts the in-vehicle computer system possibly without knowledge on the implications of the decisions. In case 2 the department for cabs ran the project, and in case 3 the department for hydraulics initiated the project.

Specification - This topic includes questions on what was specified in different stages of the project and more specifically what in terms of technical integration was specified. The three cases all reported flaws in specification, but case 2 showed that at least the basic outline of technical integration was specified.

Integration solution - The questions in this topic relate to how the technical integration was to be designed. The answers show case 1 and 3 as having higher impact on the electronic platform by requiring more information exchange. Both the system safety analysis and the closeness to product behavior work to put higher demands on technical integration in these two cases.

Verification - The topic of verification showed that a verification plan was incomplete in all three cases. For case 1 the respondents stated problems with the iterative process of verifying a software component without access to the source code and therefore having difficulties with debugging.

Result - The result in terms of project time, cost, and quality was queried in this topic. The results show that case 1 and 3 were behind schedule and had run into unforeseen problems. Case 2 was on time and cost.

Future - This topic covers ideas for improvement of integration solution and on system architecture to support integration. The result shows two problem areas. Firstly, specification problems – problems related to how to specify architecture and design principles, and secondly, method problems of assessing the embedded computer and software part of a mechatronic component. The respondents indicate that one solution is to specify system architecture demands on supplier better. Another is that the implications of system safety requirements must be included in the constraints on integration solution.

OEM Recommended Practices. From the interviews, we have analyzed problems and extracted experience from involved staff. This knowledge has been analyzed and elaborated into a list of recommended practices for integration of computer sub-systems. We support each recommendation with reasoning and findings from the study.

A development project involves many variables with no chance of sampling many data points. Instead, we base our recommendations on reasoning around the reported problems and the reported advice for improvement. We are confident that this automotive OEM has problems that are representative to OEMs in general, but the severity of each problem may well differ in different organizations. Even though, these reported problems show up as central in this study, there may be other that are more central if another set of projects were studied. Nonetheless, the recommended practices are valid to tackle the type of problems found in this study as we show by analysis here.

Thus, our list of recommendations is based on:

- Solutions suggested by industry practitioners.
- Our own analysis of the reported problems.

Recommendation 1. The integration solution for embedded computers should be decided for each candidate component in order to evaluate and compare choices.

The electronic integration solution includes decisions on how a candidate component shall fulfill system wide policies on e.g. diagnostics, fault behavior, and network management. Finding a solution to this problem involves estimating costs and negotiating against benefits.

For instance, a component may be required to signal sensor faults in a prescribed way according to the chosen architecture. Either, the supplier can adapt signaling at a cost or the OEM can decide to develop translator functionality. As we can see from this example, the available solutions quickly get technical and there is no simple way of estimating the effort of integration without deciding very precise design solutions to meet all architecture constraints.

Finding: Evaluation of mechatronic components performed in early development phases require evaluation of integration solution and thereby detailed technical details on computers and software.

Our cases show that either evaluation of supplier computer systems has not been seen as a potential problem or judged too difficult to do.

In all three cases, decisions on which component to use have been taken before decisions on integration design has been made. System engineering in early phases should include evaluation of alternatives with respect to feasibility, cost and quality. In order to do this, a specification on product functionality is needed. Since automotive products are often built from platforms, the constraints of the platform must be known i.e. the architecture description. Also, the constraints imposed by the component and the platform must be known.

Firstly, architecture constraints can be difficult to communicate in full. The architecture assumptions include issues on system states such as reduced mode, communication protocols and message identifiers, and design principles such as fault behavior.

Even in a case of complete knowledge on both architecture and component, there can be a negotiation with a supplier on what could be changed. The volume of production affects the willingness of the supplier to make a specialized design and such information may not be mature in early concept phases.

A mechatronic component may be chosen even if the electronics and software show mismatch if the benefits of the mechanical parts outweigh the drawbacks of electronic integration. In order to make informed choices in evaluating potential mechatronic components, system evaluation must include the integration solution of computer systems. In order to minimize project risks, the integration solutions should thus be specified early. Putting too little effort on specifying the integration postpones getting to know the effort involved.

An important part of the evaluation of components is to ask “what is possible to redesign?” There is a difference between buying off-the-shelf components and buying development of components. Part of the integration solution is deciding on how much of the component need changing.

Recommendation 2. All functionality should be decided prior to comparing different supplier components.

The system level functionality and all interaction between component and system should be decided. Examples are system degradation behavior, fault signaling, and production tests. The study shows that much of the focus prior to choosing component was on the functionality of the component e.g. its performance and not on system interaction issues.

Finding: The problems in integration of computer systems are mostly related to system level behavior and none of those problems can be attributed to the quality of components or the suppliers.

There were no problems reported to indicate that the components would be of low quality or not providing the desired functionality. Instead, the problem areas were all related to achieving functionality, or quality for the whole computer system such as constructing a maintainable system. These targets are achieved by system architecture choices and those cannot be foreseen by suppliers. The cases showed that there were functions in components that did not conform to platform architecture choices such as diagnostic protocol, or fault behavior, but only because they were not specified. Suppliers can and do have different quality goals than the OEM and cannot fulfill quality goals unless they are specified.

Any function that is not fully specified presents a risk of delay and increased cost. The results of our study show that problems of added functionality could, in the investigated cases, be solved although it was the OEM who got added cost.

Finding: Premature decisions on choice of component can have impact on project resource *consumption. This can stem from issues that seem minor in an early phase.*

In the early phases of development when the broad directions for physical vehicle architecture and work breakdown structure is decided, an issue like which message id's are used by the software of a supplier component can seem petty. Yet it has the potential to cause mismatch resulting in decisions of redesign later. Even though software is "soft" and therefore supposedly changeable, both the OEM and the supplier are likely to have e.g. product line constraints that prevent easy solutions.

Recommendation 3. In order to make informed choices in selecting mechatronic components, cross-functional teams should be involved including roles in electronics, service, verification and more.

To reduce project risks, all affected roles in the OEM should participate in the evaluation. The focus of this study is computer system integration and here it was evident that computer system engineers were involved late.

Finding: Many departments within an OEM other than computer system development can initiate projects which involve adding components with embedded systems to the vehicle.

In our study the problem of integrating supplier computer systems was shown to be underestimated in favor of assessing component performance. There is no simple way of predicting integration effort without specifying integration and functionality in detail.

Finding: Integration solutions can range from very minor design to large design overhauls.

The effort of integration can be substantial, or minor. In the case 2, besides that the physical constraints such as dust and EMC requirements had to be verified, the integration consisted mainly of specifying communication and then verify it. No safety, performance or real time requirements. In case 3 on the other hand, the integration required redesign of the component, adding a communication channel, or redesigning the electronic platform with e.g. a gateway.

Recommendation 4. OEMs should decide on responsibilities of different parts of the organization prior to selecting component or designing integration solution.

With lacking definitions on who owns design and who will do maintenance, there is a risk that problems will not be considered. From the study we found that different persons answered differently on questions on ownership, and maintenance responsibility.

Finding: The computer system spans the entire vehicle and it is not always evident what role is responsible for each computer subsystem.

The study shows that in a cross functional organization there can be several candidates for ownership and responsibility. For example a computer subsystem in a hydraulic application like the one in case 1 could well be owned by either one of the Electronics, Service, Hydraulics departments. Or it could be maintained by the supplier.

An automotive OEM, however, will always be responsible for the delivered product. Thus, by failing clear definitions, an OEM take the risk of having inefficient and costly service operations.

RELATED WORK

What architectures are used to enable integration? What integration solutions are used to ensure qualities? Here, we list architecture approaches in the automotive domain and describe their relation to integration.

Integration in Volvo Cars. Volvo Car Corporation, VCC, employs an electronic architecture described in [1][4]. A typical configuration of a Volvo CC car includes ECUs from more than 10 suppliers connected via several communication networks. The suppliers develop the ECUs and VCC as an OEM specifies communication, power consumption, diagnostics, and software download procedure. The volcano concept [18] allows VCC to keep a list of all signals on the network and give suppliers precise specifications on which signals to use and how much bandwidth to use. Also, worst case timing is predicted by the volcano tools. Furthermore, global states of the system are specified via state charts.

The method relies on well defined interfaces and high level specifications to suppliers. Each ECU must include a diagnostic kernel and a network interface provided by Volvo CC. Thus, the main role of VCC is to integrate components developed by suppliers.

Compared to the three cases we have studied, the approach differs as it requires buying supplier development whereas VCE purchased components off the shelf with only smaller changes feasible.

DECOS. A recent research project dealing with integration problems of large software units is the DECOS project [7]. The project has relation to integration and the objectives are to "move into component-based design … and an appropriate integration methodology … for dependable embedded real-time systems."

DECOS is targeting a broad application domain, including automotive and aerospace applications. By providing a Platform Interface Layer (PIL) and a middleware with basic services, components can be developed independently allowing for easy integration of both safety-critical and non safety-critical Distributed Application Subsystems (DAS). A DAS provides a nearly independent distributed subsystem interconnected using virtual networks. The core of DECOS is the time-triggered communication system backbone. On top of this, virtual networks are supported allowing for most types of existing networking technologies to be emulated. DECOS provides both spatial and temporal partitioning, preventing overwriting memory elements of other jobs (data and code), interference among jobs sharing access to devices, as well as the disturbance of timing among jobs holding shared resources.

AUTOSAR. The current development trends in automotive software also calls for increasing standardization of the software structure in the nodes. The upcoming automotive software standard is AUTOSAR developed by the AUTOSAR consortia [8]. AUTOSAR is scheduled to be complete in 2006, and 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 a Run-Time Environment (RTE) routing communications between software components regardless of their locations, both within a node and over networks. Tools allows for easy mapping of software onto the existing architecture of nodes (ECUs). AUTOSAR is working towards integration of standardized tools relying on, e.g., operating system standards such as, e.g., OSEK/VDX OS and communication standards as, e.g., OSEK/VDX COM [9], FlexRay [10], CAN [11], LIN [12], and MOST [13].

Software integration – CBSE approach. Component-based software engineering [16] is a software engineering approach to cost efficiently deal with software variability, reusability and maintainability. However, we claim that most CBSE related research targeting embedded systems do not address problems related to assembling applications based on large and complex commercial off-the-shelf components developed by different vendors and sub contractors [17]. The components are used mainly for dealing with reuse and product line management within a company, e.g.,[14][15]. Software components for these purposes are relatively small in comparison to the amount of functionality e.g. a control software for brakes, that is traded between different organizations in the class of automotive systems that we are focusing on.

DIRECTIONS FOR STUDIES

The study shows that the architecture of the electronic and software system in the vehicle impacts integration effort. Also, both architecture and integration solution impacts overall system qualities such as safety, reliability, and maintainability. The three cases show that quality targets were met by designing integration solutions although the effort was larger than planned. Knowing what architectures support integration would be valuable to automotive OEMs, due to the system complexity and component oriented business situation. Since qualities can impose conflicting requirements on product design, studies of relation between architecture, integration and qualities are desired to OEMs.

Finding 1 shows that an OEM should know all architectural constraints in order to minimize problems in integration. Thus it would be interesting to compare alternatives for specifying software and system architectures. The areas of model-based development and tools in system engineering could provide knowledge.

The software architectures DECOS and AUTOSAR are clearly aimed at providing platforms for development of automotive applications. Integration of software components within these frameworks should impact positively the desired qualities of automotive in-vehicle systems. In order to find architecture and integration solutions for automotive electronics and software systems, we will continue to monitor the results produced in these projects.

Techniques and methods for evaluating the impact of design choices in early phases of development are interesting to OEMs due to the problems reported in Finding 3. Large impact decisions are made in the concept development phase in terms of final system qualities.

The study shows that several systems engineering activities are important in integration and thus a suitable development process that supports activities of integration is thus desired for automotive OEMs.

CONCLUSION

In this paper, we have presented three cases of integration of mechatronic components into commercial construction equipment machines. We have focused on the integration of embedded computer systems into a vehicle electronic platform. The cases include project context, technical design as well as identified problems.

Based on the three cases we have identified problem areas in performing integration and elaborated on what causes problems in general in embedded computer and software integration. The findings are summarized below.

In the context of integration of computer and software components into automotive in-vehicle computer systems, we identify that:

Finding: Evaluation of mechatronic components performed in early development phases require evaluation of integration solution and thereby detailed technical details on computers and software.

Finding: The problems in integration of computer systems are mostly related to system level behavior and none of those problems can be attributed to the quality of components or the suppliers.

Finding: Premature decisions on choice of component can have impact on project resource consumption. This can stem from issues that seem minor in an early phase.

Finding: Many departments within an OEM other than computer system development can initiate projects with involve adding components with embedded systems to the vehicle.

Finding: Integration solutions can range from very minor design to large design overhauls.

Finding: The computer system spans the entire vehicle and it is not always evident what role is responsible for each computer subsystem

Based on these findings we provide a list of four recommended practices that would counter the problems reported. The recommendations are supported by reasoning on data from the study. The analysis shows that integration of embedded computers should be considered early in development and in order to reduce project risks, the early assessment of computers need be fairly detailed.

Moreover, we have provided a compilation of work related to quality goals of integration. Compared to the investigated cases, Volvo Cars are shown to have a method more centered on suppliers developing electronic and software components instead of off-the-shelf components. Volvo Cars focuses on system specification and integration and this decreases the problem of architectural mismatch, but architecture and specification problems are still valid.

The DECOS and AUTOSAR initiatives are aimed at software architectures for automotive applications. Both have explicit objectives to ease integration of software components and both can add in providing leverage over quality targets when integrating supplier computer components in OEM electronic platforms.

Based on the finding and the related work, we provide a set of issues that would benefit from further studies. We present areas where research would improve leverage on achieving quality targets for OEMs in integrating computer and software systems.

In summary, the challenges in achieving targeted qualities when integrating electronic components arise from issues in architecture specification and communication as well as systems engineering practices. Tackling the problems will require focus on computer system integration in early phases of development.

REFERENCES

[1] Eklund, U., Askerdahl, Granholm, Alminger, and Axelsson, Experience of Introducing Reference Architectures in the Development of Automotive Electronic Systems, 2nd Intl. ICSE workshop on

Software Engineering for Automotive Systems, 2005.

[2] Axelsson J., Towards an Improved Understanding of Humans as the Components that Implement Systems Engineering. In Proc. 12th Symposium of the International Council on Systems Engineering (INCOSE), Las Vegas, July 30-August 1, 2002.

[3] Fröberg, J., Åkerholm, M., Integration of Electronic Components in Heavy Vehicles: A Study of Integration in Three Cases, MRTC technical report.

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

[5] Leveson, N., Safeware: System Safety and Computers, Addison-Wesley, 1995.

[6] IEEE Standard 610.12-1990 Glossary of Software Engineering Terminology. 1990.

[7] DECOS, Dependable Embedded Components and Systems, homepage, https://www.decos.at/

[8] AUTOSAR, Homepage of Automotive Open System Architecture, http://www.autosar.org/

[9] OSEK/VDX, Open Systems and the Corresponding Interfaces for Automotive Electronics, http:// www.osek-vdx.org/

[10] FlexRay Consortium, Flexray, http:// www.flexray.com/

[11] ISO 11898, International Standards Organisation (ISO), Road Vehicles - Interchange of Digital Information - Controller Area Network (CAN) for High-Speed Communication, 1993.

[12] LIN Consortium, LIN - Local Interconnect Network, http://www.lin-subbus.org/

[13] MOST Cooperation, MOST - Media Oriented Systems Transport, http://www.mostcooperation.com/

[14] Hansson, H., Åkerholm, M., Crnkovic, I., Törngren, M., SaveCCM – a Component Model for Safety-Critical Real-Time Systems, Proceedings of 30th Euromicro Conference, Special Session Component Models for Dependable Systems, 2004.

[15] Lundbäck, K.L., RubusOS, Rubus OS Reference Manual -- General Concepts, Arcticus Systems: http://www.arcticus.se/

[16] Crnkovic, I., Larsson, M., Building Reliable Component-Based Software Systems, Artech House publisher, ISBN: ISBN 1-58053-327-2, 2002

[17] Åkerholm, M., Nolte, T., Möller, A., Building Distributed Embedded Systems from Large Software Components, Proceedings of the 2nd Embedded Real-Time Systems Implementation Workshop (ERTSI'05), December, 2005.

[18] L. Casparsson, A. Rajnak, K. Tindell, and P. Malmberg, Volcano a revolution in on-board communications, Volvo Technology Report. 1998.

BIOGRAPHIES

Joakim Fröberg is an Industrial Ph.D. student employed Volvo Construction Equipment Components Division where he is working with electronic system architecture guidelines and integration issues. Joakim is also a Ph.D. student at the Software engineering laboratory at Mälardalen University where he is working in the DRIVE project to study requirements and design of vehicle electronic architectures. He received his M.Sc. in Industrial Control System at Salford University 1996 and technology licentiate degree in 2004 from Mälardalen University.

His research interests include architectures of vehicle computer-based systems, but also systems and requirements engineering related to engineering of vehicle electronics.

Mikael Åkerholm is an industrial PhD student; he is half-time with MRTC/SEL and half-time at CC Systems. He received a master’s degree in computer science and engineering in 2003, and technology licentiate degree in 2005, both from Mälardalen University.

Mikaels research interests are component based software engineering, real-time, safety-critical, and embedded systems. He is participating in the SAVE-IT project, where he is working with a component technology that enables component based software engineering for safety critical vehicular systems.