Architecting Automotive Product Lines: Industrial Practice

Håkan Gustavsson[1] and Ulrik Eklund[2]

[1] Scania, Södertälje, Sweden, Hakan.Gustavsson@scania.se
[2]Volvo Car Corporation, Göteborg, Sweden, UEklund@volvocars.com

Abstract. This paper presents an in-depth view of how architects work with maintaining product line architectures in the automotive industry. The study has been performed at two internationally well-known companies, one car manufacture and one commercial vehicle manufacture. The results are based on 12 interviews with architects performed at the two companies. The study shows what effect differences such as a strong line organization or a strong project organization has on the architecting process. It also shows what consequence technical choices and business strategy have on the architecting process. Despite the differences the results are surprisingly similar with respect to the process of managing architectural changes as well as the information the architects maintain and update, especially in the light that the companies have had no direct cooperation.

Keywords: Architecting, Process, Case study, Automotive industry

1 Introduction

Software and electronics are today an important part in the development of automotive products. Experts [1] estimate that 80 percent of all future automotive innovations will be driven by electronics. Scania [2] claims that electronics in trucks and buses makes up 10-15 percent of the value and is increasing. Volvo Cars [3] estimates the value of electronics of a high-end car to 30 percent.

Architectural changes of distributed embedded systems are either evolutionary or revolutionary [4], and the architecture plays a vital role to the success of the product line. The main purpose of this paper is to understand how architecting is performed to keep up with evolutionary changes. This is summarized in the research question to be answered: What tasks are performed in the process of architecting automotive embedded systems?

Decisions in the development process [5] and within the architecting process [6] has been previously studied. Dobrica and Niemela [7] makes a comparison of eight different available software architecture analysis methods. Experience reports of introducing product lines in the automotive domain for the first time has been done previously [8] as well as showing the benefits of the introduction [9]. In a survey of 279 IT architects in the Netherlands Farenhorst et al. [10] concludes that architects are lonesome decision makers; not very willing to share architectural knowledge, but eager to consume.

This paper presents a comparison of how architects at two different companies work with maintaining existing product lines. The case study has been performed at two automotive companies, the truck and bus manufacturer Scania and the car manufacture Volvo Cars. In the next section a brief presentation is given of a general automotive electrical system. In Sec. 3 the method used in the study is presented. An outline of the case study is given in Sec. 4 followed by the results in Sec. 5. Finally we discuss the findings from our work.

2 Background

2.1 The Systems and Their Architecture

The electrical system in both cars and trucks/buses are an embedded software system consisting of 30-70 different Electronic Control Units (ECUs), each with a microprocessor executing in the order of 1 MByte compiled code[1]. These ECUs control the behavior of virtually all electrical functions, from power windows to valve timing of the engine. The in-vehicle software share a number of characteristics common to the automotive domain (see e.g. [11], [12] and [13] for further elaboration):

- A large number of vehicle models with varying feature content and configurations which must be supported by the software
- Highly distributed real-time system
- Distributed development at vehicle manufacturers and suppliers
- Low product cost margins
- Stringent dependability requirements

This combination of characteristics, together with a steady growth of features realized by electronics and software, makes the electrical system in a vehicle a highly complex software system.

Almost all ECUs have a number of sensors and actuators connected to them depending on purpose and location, and these can be shared among distributed functions. Most ECUs are reprogrammable, i.e. has flash memory and not ROM, which allows programming both in the manufacturing plant as well as at dealers and workshops after delivery to the end-user. The layout of which ECUs are connected to which bus and what ECUs are acting as communication gateways between the buses is the network topology of a vehicle, of which Fig. 1 is a representative example. The interface between the software application on each ECU is in a Scania vehicle defined by the J1939 standard [14], which is very detailed in what information is. Volvo Cars

---

[1] A few safety-critical ECUs have two microprocessors for redundancy or internal monitoring.

uses a proprietary solution for the multiplexed communication which allows a high degree of flexibility in defining and maintaining interfaces on the buses [15]. Much of the activities regarding the logical architecture at both companies are focused on these interfaces.

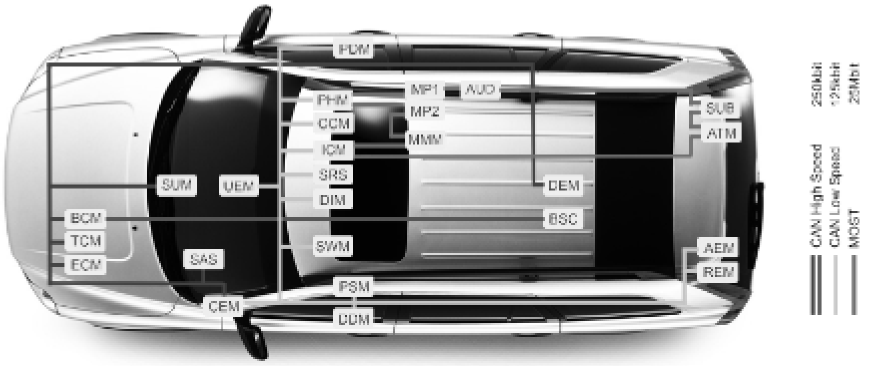

Fig. 1. The network topology of a Volvo XC90. The ECUs connected to CAN and MOST and the main multiplexed networks are seen in their approximate physical location. See [16] for a more in-depth description of the network topology of both Scania and Volvo vehicles.

2.2 Related Work

Almost all of the cases we found regarding product lines focused either on the prerequisites for a successful product line approach or the change management of an organization adapting a product line where it previous not had one. Some examples from the automotive industry are [17], [8] and [18].

Buhrdorf et al. [19] reports about the transition Salion did to a product line with a reactive approach where the necessary variations was not explored when introducing the product line, but rather handled in what they call the “steady state”. The architecting work in this paper is also reactive with the same definition, since it is about updating the systems and their architectures to comply with prerequisites not known when the architecture was first designed.

3 Methodology

The data used in this study is based on interviews with the persons most involved in the activities of maintaining architectures, i.e. the architects themselves. Neither Scania nor Volvo makes a distinction of the roles for system and software architects. All architects available and willing to participate were interviewed, which resulted in more than half of the architects at each company participating, 4 at Scania and 5 at Volvo Cars. In addition to this the managers for the architecture groups were interviewed at both companies, totaling the number of interviews to 11. Of the 11 respondents 2 were women.

The interviews were performed by the two authors, which are native to Scania and Volvo Cars respectively (see [20] for the definition of “native” in this context). One lead the interview while the other took extensive notes, which was later edited for spelling and grammar. The respondents had the possibility to read and comment the notes from their respective interview to correct any misunderstandings, purse errors or other mistakes in the recordings. This was done before the analysis took place.

The interviews were semi-structured with open-ended questions. The researchers paid special attention to not use any terminology that had special or different meanings at the two companies to avoid the respondents perceive the same question differently depending in which company they were working. After the interview was constructed, it was tested on one person at each company who had worked as a system architect to evaluate the relevance.

The interview questions were defined in English and then translated to the native language of the interviewers and respondents, Swedish, for a more natural and fluent setting. Whenever a quote from the interviews is presented in the article the translation to English was done post mortem.

The interview started with some introductory questions to get some background about the respondent, like education, professional experience of embedded systems, time employed and a general idea of how they would define architecture. The majority of each interview was based on a set of questions directed at exploring the respondent’s view of their work with the architecture. The set of questions were aimed to cover all stages of an architecting process from [21] to make sure no vital information was missed. All 11 interviews progressed in essentially the same order.

3.1 Analysis Procedure

The analysis was made by the two researchers jointly looking for common themes based on the interview questions. Also answers relating to these themes given in other questions were including in this analysis. The themes were also analyzed if they showed a close similarity between the two companies or significant differences. The two authors used their insider knowledge about respective organization and products in making the analysis and to enrich the conclusions made.

4 The Case Study

The main objective of this study was to get the richest insight possible into how architects maintain an existing architecture in practice. The selection of the two automotive companies was made for three reasons. The first is that the authors already had inside access to the subjects and the support of middle management to perform this and similar studies. Second the two companies are similar enough for a comparison to be manageable, such as each company having a product line architecture approach, but still different enough for the interviews not to be a duplicate. The third, and not least, reason is the possibility for the authors to use their knowledge as insiders to augment the analysis of the data to provide an even richer insight into the two cases.

4.1 Context

Both companies studied are situated in Sweden and share characteristics common among Swedish engineering industries such as; solid knowledge about the product among the developers, putting value on personal networks, and similar educational and demographic background in the development departments. The overall product development process at both companies follows a traditional stage-gate model. An important difference is the balance of power; Scania has a stronger line organization [22] while at Volvo Cars the project organization is stronger.

All participants had a similar educational background with an engineering master degree from a Swedish university. They had worked with embedded systems between 5 and 25 years. They also had similar experience working as architects, with a majority being an architect for 4-6 years. The main difference was that the architects at Volvo Cars had on average worked twice as long in the company, compared to Scania.

Scania is one of the world's leading manufactures of heavy commercial vehicles selling on a global market with a solid reputation of designing and producing vehicles with the core values of “Customer first”, “Respect for the individual” and “Quality”. During 20082 Scania produced 66,516 trucks and 7,277 buses. Scania is a public company with Volkswagen AG as the largest stockholder. The development of all critical parts of the product, such as engine, transmission, cabs and chassis are centralized to the research and development centre in Södertälje, Sweden.

Volvo Car Corporation is a manufacturer of premium cars with core values[3] of “safety”, “environment” and “quality”. Volvo Cars produced 374,297 vehicles in 2008[4]. Volvo Cars is a subsidiary company to Ford Motor Company (as of 2010 February 23), sharing technical solutions with other brands within FMC.

4.2 The Scania Product Line

Scania has a tradition of working with a modular product design since the early 1960's. The modular system has claimed to be the main reason why the company stayed profitable every year since 1934 [23]. The internal training program teaches the three basic corporate principles of modular thinking [24]:

1. Standardized interfaces between components
2. Well-adjusted interval steps between performance classes
3. Same customer-need pattern = same solution

---

[2] http://www.scania.com/scania-group/scania-in-brief/key-figures/

[3] http://www.volvocars.com/intl/top/about/values/pages/default.aspx

[4] http://www.volvocars.com/SiteCollectionDocuments/TopNavigation/About/Corporate/VolvoSustainability/VolvoCars_report_2008_ENG.pdf

These principles are today also applied on the electrical and electronic system, besides the traditional mechanical parts. Scania does all design work towards the product line, there is no work done towards a specific product model. A project at Scania is an addition or update to one or more modules towards a specific time when it goes into production, and there is no difference if the update is purely mechanical or includes software as well, the product line approach is identical [24]. The Scania product line uses the same architecture, as well as components, for all of its three product categories; trucks, buses and engines, seen in Fig. 2. Every sold product is customer ordered and unique which is made possible through the modular system.

The software adaptation of each product is made during production. This is done by extracting a configuration file from the manufacturing product specification, which is then downloaded onto the unique product.

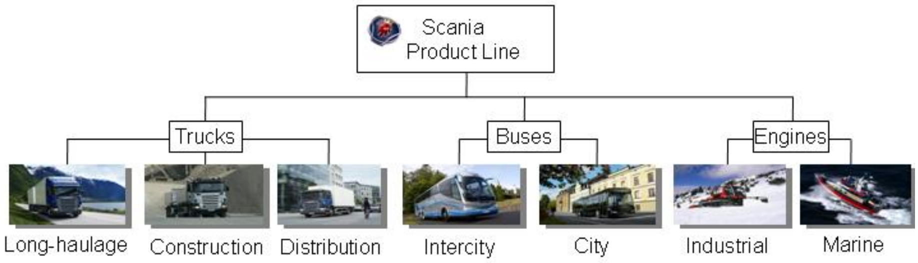

Fig. 2. The product line at Scania and the different products built on it.

4.3 The Volvo Cars Product Lines

Presently Volvo Cars maintains 3 electrical architectures for the 3 platforms in production. All vehicles in a platform are said to share the same architecture, which includes the software as well as the hardware it is executing on.

Volvo does most engineering work towards a new vehicle model, or model year, but with the intention that a solution should later be used for other vehicles on the same platform. In contrast to Scania, Volvo defines the product requirements for the individual car models and not the product line as a whole. The development of the architecture and sub-system solutions are shared between the platform and the individual products, an approach driven by the developers at the Electrical and Electronic Systems Engineering department themselves rather than a company-wide business strategy.

All vehicles produced are made to order. With the possibility for the customer to select optional features and packages the theoretical number of possible software configurations surpasses the actual number built by orders of magnitudes.

4.4 Comparison of the Product Lines

Both companies can be said to have a product line, including both hardware and software, and how they develop and maintain architectures. The electrical system share a common set of features aimed at a particular market segment, e.g. premium cars or heavy commercial vehicles, and is developed from a common set of assets (e.g. a common architecture and shared systems between vehicle models). The architectures prescribe how these shared systems interact. Since these criteria are fulfilled the software are a software product line according to [9].

The two approaches to product lines were not driven by a business decision but by the development organizations adapting to their environment. Both companies were also early adopters of the practice of building several different vehicles on the same manufacturing line, implemented years before the introduction of complex electrical systems.

Supporting factors for establishing a product line of the electrical system were in Volvo's case having a rather narrow spread in vehicle models together with an explicit single options marketing strategy (versus fixed packages). This lead to a system with a high degree of configurability. In Scania's case the supporting factors were the organization wanting to develop vehicles tailored to their customers, maximizing customer value without having to redo similar development work over and over again.

Both companies handle variability in very similar way. The architecture is predominantly implemented with two mechanisms according the taxonomy by Svahnberg et al [25]: Binary replacement—physical, where different binaries can be downloaded to the flash memory of all ECUs depending on the configuration of customer-chosen optional features such as adaptive cruise control. This can be done in the manufacturing plant using the plant's product data system with separate article numbers for software as well as hardware (including nuts and bolts) and in the aftermarket using proprietary systems. At Volvo this is accomplished by the Product Information Exchange system for software [26].

The most common variability mechanism is *Condition on variable* where all ECUs get information from a central on-board file defining the configuration of that vehicle. This file is generated automatically in the manufacturing plant and flashed as a separate binary to a central ECU. Some ECUs also store local variables similarly used in a separate binary file with its own article number as well.

5 Results

The interviews yielded results mostly regarding the process for managing an architectural change.

5.1 The Process

The process for managing changes to the architecture is very similar at the two organizations with five distinct activities:

1. need
2. impact analysis
3. solution
4. decision
5. validation

This is a fairly general process, easily mapped to a generic process for architecture work seen in Fig. 3, based on [21].

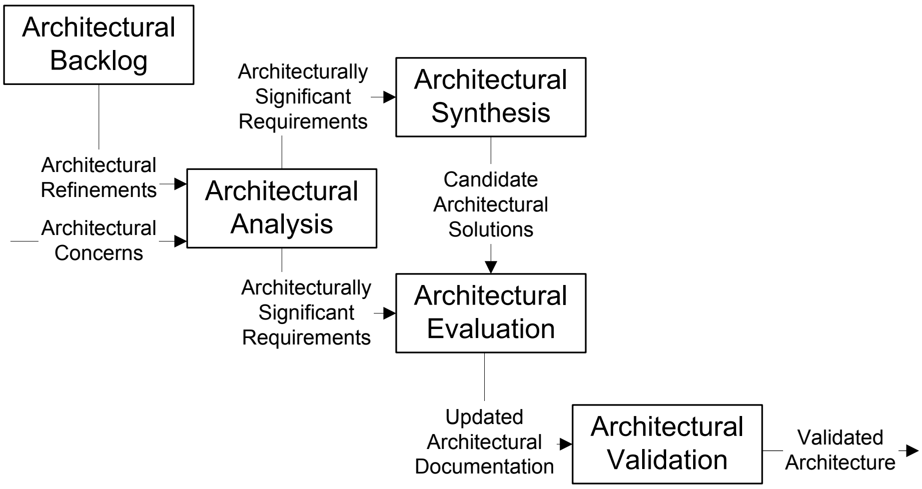

Fig. 3. A generic process for creating and maintaining an architecture, adapted from [21].

At Volvo Cars there is a greater emphasis on "why" the architecture needs to be changed, as described by one of the architects on what is done first:

> "Do a need analysis on what is driving the change. What isn't good enough? What change is needed?"

At Scania the architects' focus is on “how”, i.e. the impact of an architectural change. One possible conclusion is that the “why” is seen as a strategic responsibility of the senior architect at Scania, and the other architects are more concerned with “how”. Another possible reason is that the Scania architecting group has chosen to be more supportive than controlling.

5.2 Needs to Change the Architecture

Architects at both companies mention functional changes and functional growth as common reasons to update the architecture. This is not surprising since most new features are realized by electronics and software, and that the number of features grows almost exponentially [13].

At Volvo Cars all architects mention cost or cost reduction as a common reason to change the architecture, this is not surprising since the cost margins are very small and if an opportunity presents itself it is considered. At Scania cost was only mentioned by the manager, and then only in the context of how much the architectural change would cost. The most common reason to change the architecture mentioned by the architects at Scania was to adapt it to hardware changes, as described by one Scania architect: “Control units become too old; there is no room for development”.

5.3 Architecture Impact Analysis

The architects at Scania clearly seek to identify who is concerned by a change and what parts of the system are impacted by a proposed change. At Volvo Cars the architects request information about non-functional requirements or quality attributes and use cases when analyzing the impact, as described by one architect:

> “I need a good description of what the customer should expect form the system. If it concerns a ready solution or if it is something we should develop internally, it could be a supplier offering something which we should integrate in the system. If there is a system solution which should be integrated I want to see that as well, if there are variants and if it is to be sold as option or standard. . . “

A possible explanation to this could be that the architects at Scania are involved earlier in the development of new features or systems, while at Volvo Cars the architects are more often given a proposed technical solution, for example by a supplier. The managers at both Scania and Volvo Cars mentioned the motive for the change as important information for understanding the change, but no other architects mentioned this. We have no explanation why this is so...

The time it takes to understand the impact on the architecture from a change seems to be similar between the two companies, a few weeks to a month calendar time. It seems to depend more on finding the right stakeholders and set up appointments with them than the actual effort in man hours from the architects. Some architects at Volvo Cars also say some architectural changes takes only minutes to evaluate the impact. This could be explained by fact that such a question would not require a official

Change Request at Scania and therefore the respondents have not included these issues in their answers, or that the architects at Volvo Cars usually have a more final solution to evaluate.

5.4 Design Alternatives

Not very surprising, but notable no architecture analysis methods [7] were used or mentioned. Evaluation was in rare cases made using methods very similar to Pugh evaluation matrix [27]. Volvo architects seem to more evaluate how well different design alternatives fit into the present architecture, as mentioned by one of the architects:

> "Put some different alternatives against each other and evaluate from different aspects which is best. Cost is one example. Often the need does not come from the architecture, but from different sub-systems, from the outside. When you know what needs to be done the implementation phase begins. I follow long into the project and follow up that verification is done."

In comparison to this the Scania architects are more involved in developing different alternatives in the modelling activity. The architects see themselves as having a supporting role to function and sub-system developers. This is exemplified by

> "Requirements on new functionality are often what we start with. We then balance that against the present architecture, layout of electronics and the electrical system and weigh it against our (architectural) principles. How can we enable the functionality? Sometimes it is easy to fit in and sometimes we realize we don't have the necessary hardware and that requires a bigger effort and we go through a number of steps."

This difference in how involved the architects are in the development of subsystems is probably driven by Volvo Cars having a much larger percentage of purchased sub-systems than Scania.

5.5 Deciding on the Architectures

Architects at both companies stated that most (all?) decisions when updating the architecture were driven by non-functional requirements, quality attributes or constraints. However the attributes differed between the two companies even though the products are fairly similar, trucks/buses versus cars. The attributes deciding what update to make to the architecture could in most cases be derived from the core values for each company, for Scania Customer First, Respect for the Individual and Quality, and for Volvo Cars Safety, Environment and Quality. The attributes mentioned by Scania architects were time (to implementation), personnel resources, system utilization, including network bus load, safety, evolvability, usability, robustness, maintainability and commercial effectiveness (of which cost is a factor).

The architects at Volvo Cars unanimously mention cost as the most important factor when deciding between architectural alternatives. Other factors they mention are if the solution can realize the desired functionality, time and resources for implementation, environment friendliness exemplified by current consumption, weight, network bus load, including timing aspects, driveability, comfort and safety requirements. Risk, or minimizing the risk of a change, was also mentioned as a constraint by Volvo architects. The risk of change was not mentioned at Scania, possibly due to being obvious to think about.

A common constraint, which was mentioned by architects at both companies, was a clear wish of minimizing the effect of any architectural changes to any already existing sub-systems. The architects usually made a point of considering how a change would affect all sub-systems and not only the one proposing the change. There was a common architectural concern to have as small changes as possible, to quote one architect from Volvo Cars:

> “. . . if we need to compromise so much it hurts we have not done a good job. If we don't need to compromise so much it is good.”

5.6 Validation

The most interesting result found was that none of the architects at the two companies validated the result of the implemented change themselves. Many of the architects at Scania had a clear idea of which stakeholder they would get feedback from, the integration test group. The architects at Volvo Cars were more vague when expressing how they follow up an architectural change:

> “If it isn't a good solution we get to know there is a problem which we correct. Normally we assume that testing finds (anything).”

Common between the two companies was that the architects mentioned review of specifications on how a change in the architecture is followed up, but it is unclear exactly what documents the architects are reviewing.

5.7 The Resulting Artefacts from the Architects' Work

The resulting artefacts from the architects’ work on the changes to the architecture are very similar between Scania and Volvo Cars. It is the responsibility of the architects to update the network topology if a requested change affects how and where an ECU is connected to a network. At Volvo Cars the view of the topology is part of the officially released Architecture Description, one for each platform or product line, which is edited by the architect for the platform. At Scania the view of the topology is a separate document which is updated at every new release.

At both companies there will be a model describing the logical architecture captured in an UML tool. At Scania this model grows when a change concerns an area or function not previously modelled. Volvo Cars already has a more comprehensive model covering the complete existing system, so if the feature is not completely new it is more of a question of updating the existing model. Another artefact that gets updated is the signal database mentioned above. At Scania the architects defines message sequence charts (MSC) defining the interaction between ECUs, something that is not done at all by the architects at Volvo Cars.

The general conclusion is that the architects at both companies work with essentially the same type of information, but packaged slightly differently. Meetings are more emphasized at Scania, as stated from one of the architects;

> "…there is more eye-to-eye communication than document communication compared to other companies I have worked at."

5.8 The Timing

The timing of when a change is introduced in the architecture varies and is driven by different factors at the two companies. At Scania the most important factor mentioned is when all concerned developer stakeholders are able to update their design. All concerned developers synchronize the changes of their assets in the product line towards a common start-of-production (SOP). These change projects are tracked on visual planning boards [28].

The timing of architectural changes at Volvo Cars is usually driven by the project timing for launching new car models (also called start-of-production at Volvo Cars), or updating a new year model of an existing car. The architects respond to these change requests if they are technically possible to do within that time frame. However, in the interviews two architects expressed hesitation when claiming that it was only the project that determined the timing. To summarize: At Scania the timing of a change of the architecture is determined by the contingence of the line organization while at Volvo Cars it is determined by the need of the vehicle model project.

5.9 Other Observations

The architects at Volvo Cars had on average worked twice as long in the company, while all architects at Scania except one had worked 4 years or less at the company. The conclusion is that at Volvo Cars the architects were recruited internally form other roles while at Scania the architects were employed specifically into that role. One noticeable difference to this is the senior architect at Scania with 21 years in the company; he is also the only one of the 11 interviewees with an official recognition as senior or expert in the two organizations.

The difference in work tasks between Scania and Volvo Cars is that at Scania the architects usually works with a specific domain, e.g. HMI or chassis systems, while at

Volvo the architects were responsible for a platform and the entire system on it, e.g. the large platform (S80, V70, XC60, . . .).

6 Discussion

The striking conclusion and the answer to the stated research question is the similarity between the two companies in the tasks performed when maintaining and changing architecture. The *tasks* mentioned by the architects at both companies are virtually identical; need ⇨ impact analysis ⇨ solution ⇨ decision ⇨ validation.

The tasks do not seem to be different for architecture maintenance compared to developing a new architecture. Likewise they seem to be the same whether it is updating a product line architecture or updating the architecture of a single-shot system. Also the types of information the architects work with, one could say the viewpoints, is almost identical between the two companies. The difference being sequence charts are only used at one company but there the architects say they maintain them as a service to other stakeholders and they are not architecturally relevant. The description of the architects as lonesome decision makers made by Farenhorst et al. [10] could not be seen in this study. One possible reason for this could be the cultural differences between Sweden and the Netherlands.

The similarity in process and information is surprising since the present processes of the two companies have evolved almost independently at respective company. The similarities could be explained by the systems in cars and commercial vehicles are similar and that the companies are not too different in the demographics of their architects in terms of experience, education etc. One reason could be that the processes found can easily be mapped to a general process for architecture work, as found in [21].

As shown by Nedstam [6] there is large difference of how work is done in an organization with strong line management and a organization with strong projects. Several of the observed differences between the two companies could have affected how they work with architectural change, such as the differences in their product line approaches, the focus on project versus line organization and differences in quality attributes.

The fact that Volvo Cars has a higher degree of tool support while Scania are more conscious with respect to processes was also expected to affect the work of the architects more than was found in this study.

Acknowledgments. We would like to thank the architects and managers at the two companies for their interest and cooperation.

This work has been financially supported by the Knowledge Foundation and the Swedish Agency for Innovation Systems (VINNOVA) as part of the FFI program.

References

1. Grimm, K.: Software Technology in an Automotive Company - Major Challenges. International Conference on Software Engineering (2003) 498--503
2. Edström, A.: Hasse vill ha mer processorkraft. Elektroniktidningen (2008) 26-29
3. Edström, A.: Urban på Volvo hyllar säkerheten. Elektroniktidningen (2006)
4. Axelsson, J.: Evolutionary Architecting of Embedded Automotive Product Lines: An Industrial Case Study. In: Rick Kazman, F.O., Eltjo Poort and Judith Stafford (ed.): Joint Working IEEE/IFIP Conference on Software Architecture (WICSA) & European Conference on Software Architecture (ECSA (2009) 101-110
5. Gustavsson, H., Sterner, J.: An Industrial Case Study of Design Methodology and Decision Making for Automotive Electronics. Proceedings of the ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, New York (2008)
6. Nedstam, J.: Strategies for management of architectural change and evolution. Lund University, Department of Communication Systems, Faculty of Engineering, Lund (2005)
7. Dobrica, L., Niemela, E.: A Survey on Software Architecture Analysis Methods. IEEE Transactions on software engineering 28 (2002) 638-653
8. Steger, M., Tischer, C., Boss, B., Müller, A., Pertler, O., Stolz, W., Ferber, S.: Introducing PLA at Bosch Gasoline Systems: Experiences and Practices. Software Product Lines (2004) 34-50
9. Clements, P., Northrop, L.: Software product lines : practices and patterns. Addison-Wesley, Boston, Mass. (2001)
10. Farenhorst, R., Hoorn, J., Lago, P., Vliet, H.v.: The lonesome architect. Joint Working IEEE/IFIP Conference on Software Architecture (WICSA) & European Conference on Software Architecture (ECSA). IEEE (2009) 61-70
11. Schulte-Coerne, V., Thums, A., Quante, J.: Challenges in Reengineering Automotive Software. IEEE Computer Society, Kaiserslautern, Germany (2009) 315-316
12. Pretschner, A., Broy, M., Kruger, I.H., Stauner, T.: Software Engineering for Automotive Systems: A Roadmap. International Conference on Software Engineering (2007) 55-71
13. Broy, M.: Challenges in automotive software engineering. Proceedings of the 28th international conference on Software engineering. ACM, Shanghai, China (2006) 55-71
14. SAE: Standard J1939 - Recommended Practice for a Serial Control and Communications Vehicle Network. Society of Automotive Engineers (2009)
15. Casparsson, L., Rajnak, A., Tindell, K., Malmberg, P.: Volcano-a revolution in on-board communications. Volvo Technology Report, Vol. 1 (1998) 9-19
16. IEEE-1471: IEEE Recommended practice for architectural description of software-intensive systems. IEEE Std 1471-2000 (2000)
17. Voget, S., Becker, M.: Establishing a software product line in an immature domain. Software Product Lines, Vol. 2379. Springer (2002) 121-168

18. Tischer, C., Muller, A., Ketterer, M., Geyer, L.: Why does it take that long? Establishing Product Lines in the Automotive Domain. 11th International Software Product Line Conference, Kyoto, Japan (2007) 269-274
19. Buhrdorf, R., Churchett, D., Krueger, C.: Salion's Experience with a Reactive Software Product Line Approach. Software Product-Family Engineering (2004) 317-322
20. Brannick, T., Coghlan, D.: In Defense of Being" Native": The Case for Insider Academic Research. Organizational Research Methods 10 (2007) 59
21. Hofmeister, C., Kruchten, P., Nord, R.L., Obbink, H., Ran, A., America, P.: Generalizing a Model of Software Architecture Design from Five Industrial Approaches. Proceedings of the 5th Working IEEE/IFIP Conference on Software Architecture. IEEE Computer Society (2005) 77-88
22. Bergsjö, D., Almefelt, L.: Supporting requirements management in embedded systems development in a lean influenced Proceedings of International Conference on Engineering Design, Dubrovnik, Croatia (2010)
23. Johnson, H.T., Senge, P.M., Bröms, A.: Profit beyond measure : extraordinary results through attention to work and people. Nicholas Brealey, London (2000)
24. Kratochvíl, M., Carson, C.: Growing modular : mass customization of complex products, services and software. Springer, Berlin ; (2005)
25. Svahnberg, M., Van Gurp, J., Bosch, J.: A taxonomy of variability realization techniques. Software: Practice and Experience 35 (2005) 705-754
26. Melin, K.: Volvo S80: Electrical system of the future. Volvo Technology Report, Vol. 1 (1998) 3-7
27. Pugh, S.: Total design : integrated methods for successful product engineering. Addison-Wesley, Wokingham (1990)
28. Morgan, J.M., Liker, J.K.: The Toyota product development system : integrating people, process, and technology. Productivity Press, New York (2006)