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

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

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

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

Abstract

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

1. Introduction

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

1.1. Integration in automotive products

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

1.2. Problem description

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

1.3. Our proposed method

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

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

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

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

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

2. Vehicle electronic systems

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

2.1. General properties

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

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

2.2. Integration strategies

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

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

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

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

I1. New ECU connected directly on a system bus.

I2. New ECU connected via a gateway.

I3. Application software component located in existing ECU.

I4. New ECU stand alone - not connected to a bus.

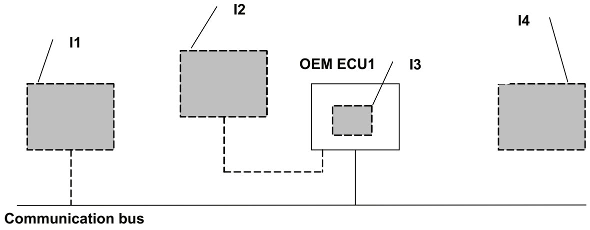

Figure 1. Four choices in integration strategy

2.3. Example: Gearbox

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

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

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

3. The method explained

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

3.1. ATAM

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

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

3.2. AHP

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

Table 1. Element comparison

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

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

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

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

Table 2. Algorithms used in chainwise paired comparison

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

estimates are consistent.

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

Full consistency can be hard to achieve in practice with many factors to chainwise compare. To compensate for this inconsistency we compute a new estimated ratio, $\tilde{R}_i$, with Equation 3. $\tilde{R}_i$ is by definition a consistent estimation, fulfilling Equation 2.

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

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

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

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

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

3.3. The proposed method

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

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

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

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

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

4. Using the method

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

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

4.1. Scenarios

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

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

Safety

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

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

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

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

Reliability

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

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

S7. Minimum number of connectors wanted

S8. Testable design wanted

S9. Simpleness preferred

S10. Fault diagnosis desired

Modifiability

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

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

S13. Porting SW platform to new hardware

Serviceability

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

S15. Physical components are easily replaced

S16. Software functionality is easily replaced

4.2. Prioritizing the scenarios with chain-wise paired comparison

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

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

Table 3. Scenarios prioritized with chainwise paired comparison

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

Table 4. Scenario S8

i	R_i	D_i	I_i	$\tilde{R}_i$	M_i	V_i
1	I_1/I_2	2	2,400	2,093	2,866	0,274
2	I_2/I_3	$\frac{1}{4}$	0,300	0,262	1,369	0,131
3	I_3/I_4	5	6,000	5,233	5,233	0,500
4	I_4/I_5	$\frac{1}{3}$	0,400	0,349	1,000	0,096

Table 5. Decision matrix

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

4.3. Weighting scenarios against an integration strategy

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

5. Analysis

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

5.1. The method compared to AHP and ATAM

The method does provide a structured way of using expert knowledge to make decisions in design of automotive electronics and possibly many other areas. Like ATAM recognises, the difficulties in making decisions stems from the complexity where many stakeholders have different goals. What ATAM lacks is the actual support for decision making. ATAM is instead intended to identify sensitive design points in the system, but choosing a design alternative must be done by other means. AHP on the other hand is a method for decision making with multiple criteria, but lacks a structured way of listing the criteria. Thus, using the concept of scenarios and utility trees from ATAM as input to an AHP process gives us a method that includes both benefits. Compared to using ATAM alone, the combined method supports decision making and should still have the benefits that has been reported with ATAM. One such important benefit is that stakeholders get to reason about qualities and their fulfilment. Thus, compared to using AHP alone, we will get both a structure for the criteria and likely also the benefit of stakeholder involvement and communication.

5.2. Methods pros and cons

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

A large effort in weighting scenarios could be accepted if there are many decisions to make.

- **Flexible and scalable**. As we progress through the method we can choose to employ more or less rigorous comparisons depending on the importance of the design decision. For instance it may be justified to employ the full comparison scheme as opposed to the chainwise, if we would want to integrate a new engine system with high impact on system behaviour. Likewise we can choose to have a high number of scenarios if the decision is judged very important.
- **Feedback on accuracy**. The AHP calculations produce a measure of consistency for the estimations made by the experts. Thus, both in the second and third step we will get feedback on whether the interviews have been successful. If the consistency is too low, we can instead decide to redo some of the importance assessments.
- **The method has some support for answering why**. An important issue when designing systems is to have an understanding by all involved why a certain design has been chosen. If the ”why” is clearly understood, we run a low risk that the decision is overrun by a new decision. It is clearly visible in the AHP process how the relative importance measures have been estimated. This would likely aid in the effort of explaining why decisions have been made.

6. Conclusions

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

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

References

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

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

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

[4] J. Fröberg and M. Åkerholm. Integration of electronic components in heavy vehicles: A study of integration in three cases. In *Proceedings from Systems Engineering/Test and Evaluation Conference, Melbourne, 25-27 September 2006*, Melbourne, September 2006.

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

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

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

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

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

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

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