INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 03 STOCKHOLM, AUGUST 19-21, 2003

Relational reasoning supported by quantitative methods for product modularization

Ola Larses and Michael Blackenfelt

Abstract

Modularized products potentially bring a range of benefits and consequently the subject has received much attention. Modularity has been defined and methods for modularization are developed based on two main viewpoints; either the product with its elements and relations or the purpose of the modularization has been in focus. Here is the combination of the viewpoints, referred to as the strategic and the technical viewpoint, emphasized. This is done by employing methods such as DSM and MIM. A mechatronic product that mainly is based on software is used as a case study, and it is argued that the methods used are especially suited for this type of product. It is also shown how metrics and algorithms may be used to support the qualitative decision-making.

Keywords: Architecture, modularization, mechatronics, Design Structure Matrix

1 Introduction

Modular products have received a lot of attention during the last decades, because modularity is credited with a range of potential benefits for the companies that employ its principals. It is claimed that a modular product provides better handling of product variety, improved organization of development and production as well as improved handling of various after sales issues, in comparison to a product that exhibits a lower degree of modularity.

The paper sets out to explore product modularization for mechatronics, and more specifically in the context of the hardware and software architecture of modern trucks. In the area of engineering design many previous studies have focused on the modularization of the mechanical domain. This domain is difficult to modularize because of physical limitations. On the other hand, in future systems complexity will be transferred from the mechanical domain to software and electronics, for example by so-called “x-by-wire” concepts, which will make the mechanical parts more flexible and thus easier to modularize. Moreover, it should be noted that already today in the automotive industry function growth takes place in software and electronics rather than in the mechanics. Therefore, it is highly relevant to study how to modularize the domain of mechatronics, in order to handle the increasing complexity.

In the literature various theories to describe modular products have been suggested in order to confront the question: What is a modular product? Often a modular product is described as a product with exchangeable elements for creation of variety; the modularity is described in terms of the purpose of the modularization [1]. Others describe the modularity in terms of the product elements; it is postulated that there should be a simple mapping between functions and parts, and there should be strong links between parts within the modules and weaker links between parts of different modules [2]. This paper builds on thoughts of Blackenfelt [3], who argues that both the purpose of the modularity and the products elements and relations are needed to define the architecture.

Moreover a wide range of methods and procedures for product modularization has been suggested to answer the question: How to create a modular product? It is obviously impossible to modularize a product without knowing what modularity is, and thus methods are often derived from a definition aiming either at the purpose of the modularization or the encapsulation of the product elements. The Design Structure Matrix (DSM) has frequently been employed to describe the relations between elements of the product in order to define suitable modules [4]. On the other hand, the Module Indication Matrix (MIM) with the module drivers of Erixon [5] has proved interesting for describing the purpose of the modularization. These approaches have been combined by Blackenfelt [3] and are further developed and exemplified here.

Furthermore, in the previous studies of mechanical products, using for example DSM, the module boundaries have merely been defined for an already designed product, i.e. only smaller lay-out changes have been possible when the needed parts are known. When studying products, which mainly are realized by software, there is more freedom to structure late in the process, at the point when the necessary parts are known. Thus, DSM should be more suited to be used for this type of product.

In reality, it is rare that an architecture is created without constraints, from legacy and taken decisions. In this study a new function will be mapped to an existing architecture, but the architecture of the function may be freely designed within the given constraints. These constraints will be modeled together with the product elements and the modularization purposes. To keep down the complexity of the study, only one function of the truck is considered. For the implementation of this function both new hardware and software is needed; new and existing elements need to be grouped to modules. The role of the existing and new hardware need to be defined and the new software need to be mapped onto the hardware.

2 Case study background

Much of the functionality of a car or truck today is based on electronics. The system architecture of automotive electronics has evolved over time. The first systems were simple enough to be completely understood by a single engineer, which was still the case in the 1950s. In the 1990s the complexity had grown substantially and software based electronic control units (ECU) had made an entry. The Mercedes S-class from 1991 included more than 50 ECUs and more than 3 km of wiring; newer models are even more complex.

After the introduction of software based functionality the development of a new function usually still introduced a new ECU dedicated to the purpose, which allows good modularization. The removal of a function is performed by removing a specific piece of hardware, which includes software, and then some adjustments in related components are performed. This approach is however no longer unproblematic as the relationships between components have become very complex due to the increased number of functions and interdependencies, the method is also expensive because of the excessive hardware. Today functions generally share the hardware space on ECUs and discussions on generic architectures exist in the automotive industry. The intention is to simplify the hardware structuring by handling dependencies in the structuring of the software. The solution to the modularization problem of electronics is however not yet final.

With the sharing of hardware new aspects of modularization need to be considered. When the hardware dependency of software is reduced it becomes necessary to map the software onto the existing hardware units. This mapping should be based on the requirements from a function structure. The function structure in this case study is defined by the control design. The control design implies requirements on sensors and actuators and also provides a functional decomposition that can be used for the control software. Requirements on the mapping of software to hardware can be derived, e.g. if two software blocks are closely related in the software architecture there may be timing requirements on the implementation. In a real-time control system important requirements include worst case response time, period time and communication delay.

With a defined control and software architecture the software need to be deployed on the hardware. In this process two software blocks can be separated or placed in the vicinity of each other. It is possible to distinguish between three levels of software proximity. Two pieces of software can be on the same network separated by a gateway, the same bus or in the same ECU. Software can be separated because of integrity issues; separating software reduces the risk for data contamination. Arguments for increasing the proximity are performance arguments related to communication needs. A high proximity reduces communication delays and also supplies higher bandwidth, there is significant difference between internal ECU communication and communication between different buses over a gateway.

This paper considers functional blocks from the control design and suggests a methodology to cluster these blocks. The clustering would implicate how the software should be deployed on the hardware, and also give hints for the hardware architecture.

2.1 The Scania architecture

The Scania system architecture is based on the controller area network (CAN) protocol, and it is based on three buses separated by a control unit that acts as a gateway. The J-1939 standard prescribes a set of messages that are used for the communication in the network. The gateway unit is called the coordinator and features some software functionality apart from the role of a gateway. ECUs with different levels of system criticality are separated by being placed on different buses which are identified by color labels, figure 1. The red bus has ECUs with the highest criticality. ECUs on the yellow bus are estimated to have intermediate criticality and the green bus have the lowest level of criticality.

A group of related functions are labeled as a logical ECU. Logical ECUs are then allocated on the physical ECUs. Not all control units are included for a given vehicle as the functionality varies with each truck. The architectural concept of Scania has separated much of the functionality onto separate physical ECUs but there are exceptions, for example the coordinator collects several functions. Moreover it should be noted that the main software of a given function may be located in one ECU, while the other ECUs are necessary parts of the function. Typically, the main function may be located in one ECU while the other ECUs are needed as actuators and sensors.

The ECUs are generally placed on the bus with appropriate criticality given by their functionality, but there are exceptions. An ECU might be placed on a bus with higher integrity level in order to reduce excessive communication or construction penalties associated with placing it on the bus with appropriate level of integrity. Aspects for the system design include: available technology, reliability, safety, cost, sub-contractors etc. These trade-offs are currently performed through qualitative investigation efforts. The choice to place multiple logical ECUs in one physical ECU is due to considerations in line with the trade-offs.

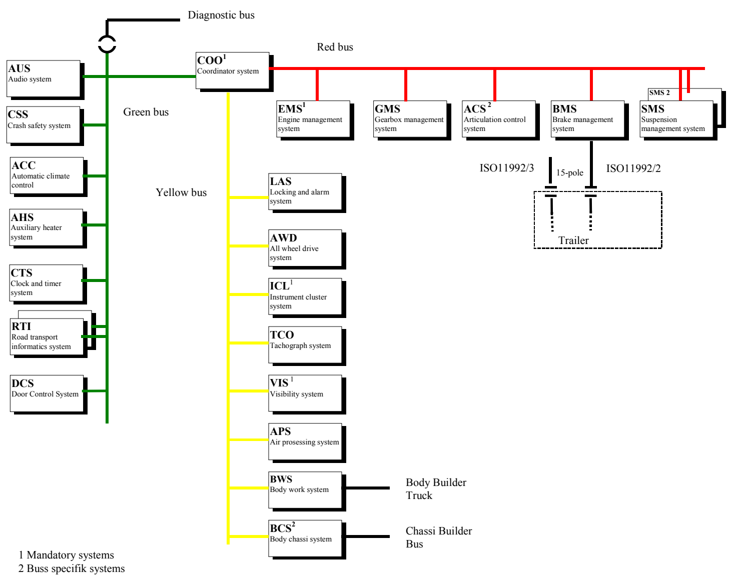
Diagnostic bus
Red bus
Green bus
Yellow bus
AUS Audio system
CSS Crash safety system
ACC Automatic climate control
AHS Auxiliary heater system
CTS Clock and timer system
RTI Road transport informatics system
DCS Door Control System
COO1 Coordinator system
EMS1 Engine management system
GMS Gearbox management system
ACS2 Articulation control system
BMS Brake management system
SMS 2
SMS Suspension management system
ISO11992/3
15-pole
ISO11992/2
Trailer
LAS Locking and alarm system
AWD All wheel drive system
ICL1 Instrument cluster system
TCO Tachograph system
VIS1 Visibility system
APS Air prosessing system
BWS Body work system
Body Builder Truck
BCS2 Body chassi system
Chassi Builder Bus
1 Mandatory systems
2 Buss specifik systems

Figure 1. Scania ECU topology.

2.2 Adaptive Cruise Control and implementation considerations

Adaptive cruise control (ACC) may be seen as an extension to the conventional cruise control, where ACC not only keeps the speed but also ensures a given distance to the vehicles ahead, by using the brakes. The ACC is mainly seen as a comfort oriented function, although it could be seen as the first step towards a more autonomous driving. In the future this step could be followed by various functions aimed at comfort, safety and fuel economy.

For ACC, some device to measure inter-vehicle speed and distance, typically radar or laser, is needed. These devices often include an ECU powerful enough for both signal processing and ACC controllers. Software for the longitudinal control may thus be placed in this unit or in any other ECU of the network. Some of the required functions and ECUs exist already in a modern truck. ECUs such as the brake management system (BMS), the engine management system (EMS) and the gearbox management system (GMS) are necessary. A yaw rate sensor is available in the BMS but a dedicated sensor could be introduced. Vehicle speed signal may be received from various available sensors.

3 Theoretical perspectives

In order to modularize the product a model, which describes the products elements and relations, is needed. Here a functional model, as described in [1], where the relations could typically be information, energy and material transfer, is used. The functions could in theory be realized in any of the mechatronic domains. It should be noted that the ECUs of the network in the current Scania architecture are not represented as elements, because they will rather carry the architecture and contain the functions and the relations. The functions should be grouped to modules, basically logical ECUs, which then may be mapped to the hardware ECUs.

3.1 Functions and technical relations

The technical relations in our system are based on the control requirements, figure 2. As most of the functional elements are implemented in software only the information transfer relation is relevant and it is given three levels of strength. Data that influence the control but are not a part of a closed loop are seen as the least strong relation. Closed control loop data relations are stronger. Closed loop control data with a high frequency are regarded as the strongest binding, while low frequency data is regarded an intermediate binding.

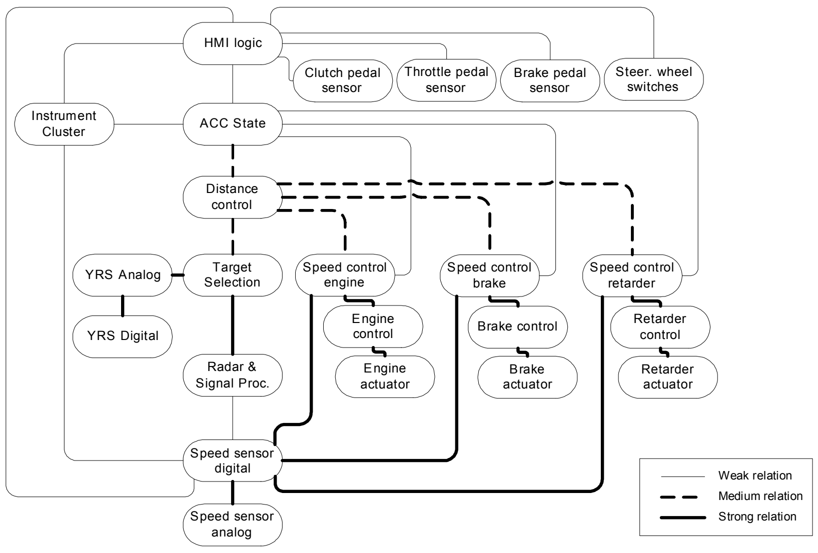

Figure 2. Technical relations based on control issues. The thickness of the lines indicates the strength of the technical relations.

3.2 Strategic relations and constraints

The strategic relations between the functions are derived from module drivers or the strategic intention for the functions, which describe the purpose of the modularization. Typically the module driver for a function could be that it should be out-sourced (buy) or developed in-house (make). When two functions both have the module driver "make" it could be argued that they strategically are similar and it would therefore be beneficial to place them within the same module. Also, it is here argued that it would be bad to place a "make" function together with a "buy" function within one module. The same reasoning may be applied to the other three couples of strategic intentions "reuse vs. develop", "slow change vs. quick change" and "commonality vs. variety", as illustrated in figure 3.

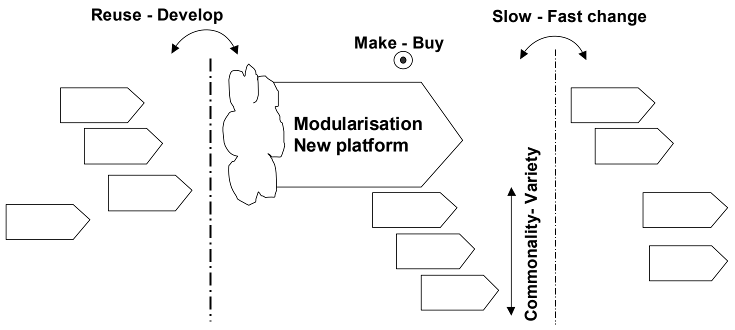

Figure 3. Strategic module drivers.

Of course there may exist other strategic intentions, but these eight (four couples) certainly are important in this case and probably in a general context for system architecture decisions. A part in the product model will be attributed the label "reuse", "develop" or possibly is no statement made for the module driver couple. Thus, the part will relate to all other parts where a statement is done for the module driver couple and therefore due to legacy it is likely that there will be a higher number of strategic relations than there will be technical relations.

The module driver couples are used to model issues such as the fact that GMS and EMS functions are traditionally developed in-house whereas BMS functions are mainly developed by the suppliers. It also considers that there exist a number of engine variants whereas there only exists one retarder. Moreover, radar sensors and radar signal processing along with target selection are hardly technologies within Scania's core competence and thus will be bought. Table 1 exemplifies module drivers. Here, the decisions about a module driver may still not be taken, e.g. make/buy for the ACC state machine, decisions that later may be supported by the clustering based on the other module drivers.

Table 1. Module drivers for selected elements.

Function	**Reuse/Develop**	**Slow/Fast change**	**Make/Buy**	**Comm./Variety**
ACC state machine	Develop			Comm.
Target selection		Fast	Buy	Comm.
Engine control	Reuse	Fast	Make	Variety
YRS, digital	Reuse	Slow	Buy	Comm

In a situation when a new function should be added to an available architecture a range of decisions are difficult to change and will remain fixed. For example, the control algorithms of the brakes will still be placed within the BMS and the brake actuators will still be controlled by the BMS. Since the ECU:s are not represented in the product model the hard constraint is e.g.: the brake actuator and the actuator control should be kept together within the same module. Thus, on top of the four couples of strategic intentions, the constraints of the available architecture and already taken decisions are added.

3.3 Relational reasoning supported by quantitative DSM analysis

The product model described by parts and their relations may be represented by a design structure matrix (DSM), to visualize the model. The relations are of three types, technical (real-time control) relation, strategic relations and constraints. In order to support the modularization the relations may be quantified based on the strength of relations [4].

The strategic relations use the same value range as [4], from -2 to +2, i.e. between a part which definitely should be out-sourced and a part that definitely needs to be developed in-house would lead to an "-2" relation. Since there are four module-driver couples the range will in total be -8 to +8.

For the technical relations the strength is based on the real-time control requirements, which actually give a reasonable indication of how the parts ought to be grouped. Here three levels are identified with the values 3, 6 and 9 to indicate the differences in requirements. These steps are used rather than 1, 2 and 3 because an initial thought is that the technical relations should weigh roughly as much as the strategic relations totally. How the technical relations should be weighed against the strategic relations of course is a delicate issue, which will be further touched upon. The direction of the real-time control communication is not regarded and if the requirements of opposite directions differ, the toughest real-time requirement set the level. No negative relations are identified due to the nature of the studied product.

The constraints are not given any value since they should be seen as hard decisions, which cannot be negotiated.

When using the DSM for clustering parts to modules the idea is that strong and positive relations preferably should be kept within the module, whereas weak or negative relations should be kept between the modules. In order to describe how well a given set of modules satisfies these principles, metrics, such as Module Independence (MI) [6] and Average Ratio of Potential (ARP), may be introduced.

$$\mathrm{MI} = \sum_{mi=1}^{n} \frac{in_{mi}}{tot} \tag{1}$$

$$\mathrm{ARP} = \sum_{mi=1}^{n} \left(\frac{in_{mi} / pot_{mi}}{n} \right) \tag{2}$$

In the equations the index *mi* identifies a specific module, *n* is the number of modules, *in* is the sum of relations within a module and *tot* is the sum of all relations in the DSM. The value of *pot* is given by multiplying the number of relations within the module with a potential maximum relation score i.e. 17 (4x2 + 9) in our case. The main difference between the two metrics is that the ARP favors a higher number of smaller modules than the MI. In case there are only positive relations the MI would suggest one big module whereas the ARP differentiates between weak and strong relations. However, for a single element module the variable *in* is equal to zero, thus clustering of elements is encouraged.

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	Instrument cluster	■	6	7	7	4	4	2	-4	-2	-2	0	0	0	-2	-2	-2	-2	1	-2	2	2	0	0
2	Steering wheel switches	6	■	7	4	4	4	2	-4	-2	-2	0	0	0	-2	-2	-2	-2	-2	-2	2	2	0	0
3	HMI statemach.	7	7	■	7	4	4	4	-4	-4	-4	3	3	3	0	0	0	0	3	0	0	0	2	2
4	ACC statemach.	7	4	7	■	10	7	7	-1	-4	-4	0	0	0	0	0	0	0	0	0	0	0	8	2
5	Distance control, overall	4	4	4	10	■	10	10	2	-4	-4	0	0	0	0	0	0	0	0	0	0	0	8	2
6	Speed control, retarder	4	4	4	7	10	■	4	-4	-4	-4	0	0	0	0	0	0	0	9	0	9	0	2	2
7	Speed control, wheel brake	2	2	4	7	10	4	■	-4	-6	-6	0	0	0	11	2	2	2	11	2	-2	-2	4	4
8	Speed control, engine	-4	-4	-4	-1	2	-4	-4	■	13	4	0	0	0	0	0	0	0	9	0	0	0	-2	-2
9	Engine control	-2	-2	-4	-4	-4	-4	-6	13	■	17	-2	-2	-2	-4	-4	-4	-4	-2	-2	0	0	-2	-2
10	Engine actuators	-2	-2	-4	-4	-4	-4	-6	4	17	■	-2	-2	-2	-4	-4	-4	-4	-2	-2	0	0	-2	-2
11	Brake pedal sensor	0	0	3	0	0	0	0	0	-2	-2	■	6	6	6	6	6	6	4	4	6	6	0	0
12	Acc. pedal sensor	0	0	3	0	0	0	0	0	-2	-2	6	■	6	6	6	6	6	4	4	6	6	0	0
13	Clutch pedal sensor	0	0	3	0	0	0	0	0	-2	-2	6	6	■	6	6	6	6	4	4	6	6	0	0
14	Wheel brake control	-2	-2	0	0	0	0	11	0	-4	-4	6	6	6	■	17	8	8	6	6	4	4	2	2
15	Wheel brake, actuators	-2	-2	0	0	0	0	2	0	-4	-4	6	6	6	17	■	8	8	6	6	4	4	2	2
16	YRS digital	-2	-2	0	0	0	0	2	0	-4	-4	6	6	6	8	8	■	17	6	6	4	4	11	2
17	YRS analog	-2	-2	0	0	0	0	2	0	-4	-4	6	6	6	8	8	17	■	6	6	4	4	2	2
18	Speed sensor, digital	1	-2	3	0	0	9	11	9	-2	-2	4	4	4	6	6	6	6	■	15	2	2	7	4
19	Speed sensor, analog	-2	-2	0	0	0	0	2	0	-2	-2	4	4	4	6	6	6	6	15	■	2	2	4	4
20	Retarder control	2	2	0	0	0	9	-2	0	0	0	6	6	6	4	4	4	4	2	2	■	17	-2	-2
21	Retarder actuators	2	2	0	0	0	0	-2	0	0	0	6	6	6	4	4	4	4	2	2	17	■	-2	-2
22	Target selection	0	0	2	8	8	2	4	-2	-2	-2	0	0	0	2	2	11	2	7	4	-2	-2	■	15
23	Radar & signal proc.	0	0	2	2	2	2	4	-2	-2	-2	0	0	0	2	2	2	2	4	4	-2	-2	15	■

Figure 4. DSM matrix with the sum of all technical and strategic relations included. The green labels points on the parts that must be kept together as a hard constraint. The different coloured borders represent three clusterings. Black (ARP=0.66), Black/Red (ARP=0.66) and Black/Red/Blue (ARP=0.74).

In figure 4 a DSM is presented with the sum of all relations in the cells. Qualitatively, it is quickly seen that the engine functions including speed control of the engine primarily have negative relations to most other functions and thus should be kept as a separate module. Speed control of the retarder and the wheel brake on the other hand fits well with the ACC distance control and the state machine. The radar including target selection may be treated as a separate module although there are positive links to the ACC distance control and state machine. Qualitative reasoning like this may then be supported by the metrics such as ARP. For example, if the big cluster with functions 1-7 should be broken up, a better ARP is actually achieved by breaking it into 1-3 and 4-7, which makes sense.

3.4 Relational reasoning supported by quantitative clustering algorithms and metrics

In the search for good modules the solution space can become extremely large, as it is possible to vary both the number and size of clusters. It should also be noted that a DSM, as in figure 4, represents only one possible arrangement of rows and columns. This makes it worthwhile to use some kind of clustering algorithms to support the search for modules. With a set of relation variables and a defined DSM with reasonable weights on relations, it is possible to treat the recorded values in a strict mathematical way. Using mathematics and selected measures, optimization algorithms can be utilized. One branch of optimization theory that considers similar problems is graph theory.

There are several different algorithms within graph theory that solves or approximate solutions to problems of clustering graphs with different constraints. Many of the problems are NP-complete. Some algorithms are implemented and readily available in tools. We have used a tool for graph partitioning called METIS [7] to establish an initial clustering. The METIS tool algorithm has some obvious differences to the problem we aim to solve. It does not take into account that the internal bindings of the cluster should be maximized and it also tries to build partitionings of equal sizes, which is not a constraint in our problem. The partitioning is still useful as an initial guess for further refinement. Algorithms implemented in Matlab have been used for refining the results. The algorithm changes the clustering to improve the MI score of the system as much as possible in every step. Moving one or two components between clusters is considered as well as switching components between clusters. The limited implementation is simple and fast and can be improved but our effort does not aim to develop algorithms to find optimal solutions, instead the results are only used as guidelines for further design. In the end the results were inspected for improvements by hand.

It is possible to create different solutions by adding an *offset* to the relations. As the algorithm is based on the MI measure, high average scores make the clusters larger and a negative offset creates smaller clusters. The choice of offset is a parameter that can be tuned to serve different purposes, for example finding clusters with different levels of strength. The different clusters derived by the described techniques were analyzed with the ARP and MI measures.

4 Results and discussion

The ACC function was broken down into 23 function elements. Both technical and strategic relations were modeled following the ideas above. Here three DSMs have been analyzed; a combined matrix with both strategic and technical relations, a matrix only with the strategic relations and a matrix only with technical relations. In the analysis a range of offsets was tested and the MI and ARP measures were calculated for each clustering. The different clusters are summarized in figure 5.

The first column is given by METIS. It can be noted that this initial clustering resembles some of the final solutions. It is also interesting to note how the results evolve with different levels of offset. The higher offset, the more clusters are found, this is expected as the technical relations are only indicated with positive values. Low offset indicates weak grouping, as the offset increase clusters become smaller but indicating stronger grouping; compare the reasoning about MI and ARP above. The strong relationships indicate good modules and that the pieces of software should be placed on the same ECU. Intermediate strength relationships should be placed on the same bus and weak relationships can be placed anywhere in the network.

The different raw clusterings give good hints for the final manual clustering, all shown in figure 5. It is noticeable that the strategic and technical clusters differ significantly. The technical relations cluster all the control software together and put the YRS and radar together, which suggests that the YRS in the BMS should not be used. The strategic relations make larger clusters, but interestingly enough some clusters coincide with the technical. The two aspects are also evaluated together and manually edited to solutions with high ARP scores. The clusters with high ARP seem reasonable, indicating that ARP is a useful metric.

Clusterset:	**METIS**	**Algorithm raw clusterings**						**Manual**				**Strat only raw**				**Tech only raw**		
Component Offset:		**0**	**2**	**3**	**4**	**5**	**6**	Low	Med	High	Max	**0**	**2**	**4**	**5**	**1**	**2**	**3**
Instrument cluster	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Steering wheel switches	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	10	10
HMI statemach.	1	1	1	1	1	1	1	1	1	1	1	1	1	1	12	1	1	1
ACC statemach.	1	1	1	2	2	2	2	1	2	2	1	1	1	1	13	2	2	2
Distance control, overall	2	1	1	2	2	2	2	1	2	2	1	1	1	2	2	2	2	2
Speed control, wheel brake	2	1	1	2	2	2	2	1	2	2	1	1	1	2	16	2	2	2
Speed control, retarder	6	1	1	2	2	2	2	1	2	2	1	1	1	0	0	2	2	6
Speed control, engine	3	3	3	3	3	3	3	3	3	3	3	3	3	3	15	2	2	3
Engine control	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Engine actuators	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Brake pedal sensor	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	10	11	11
Acc. pedal sensor	10	4	4	4	4	4	10	4	4	4	4	4	4	4	4	11	12	12
Clutch pedal sensor	7	4	4	4	4	4	10	4	4	4	4	4	4	4	4	12	13	13
Wheel brake control	4	4	4	4	4	4	4	4	8	8	8	4	4	4	4	8	8	8
Wheel brake, actuators	4	4	4	4	4	4	4	4	8	8	8	4	4	4	4	8	8	8
YRS digital	9	4	4	4	4	9	9	4	8	9	9	4	4	4	4	5	5	9
YRS analog	9	4	4	4	4	9	9	4	8	9	9	4	4	4	4	5	5	9
Speed sensor, digital	6	4	4	4	6	6	6	4	6	6	6	4	4	4	4	2	2	6
Speed sensor, analog	6	4	4	4	6	6	6	4	6	6	6	4	4	4	4	2	6	6
Retarder control	7	4	4	7	7	7	7	7	7	7	7	4	4	4	7	7	7	7
Retarder actuators	7	4	4	7	7	7	7	7	7	7	7	4	4	4	7	7	7	7
Target selection	5	1	5	5	5	5	5	5	5	5	5	1	5	5	5	5	5	5
Radar & signal proc.	5	1	5	5	5	5	5	5	5	5	5	1	5	5	5	5	5	5
ARP	***57%***	***43%***	***58%***	***65%***	***68%***	***72%***	***70%***	***66%***	***66%***	***74%***	***76%***	***44%***	***57%***	***48%***	***33%***	***46%***	***39%***	***50%***
ARPtot	*6%*	*11%*	*9%*	*7%*	*7%*	*7%*	*7%*	*8%*	*7%*	*6%*	*6%*	*11%*	*9%*	*7%*	*4%*	*4%*	*3%*	*5%*
MI pos	*30%*	*81%*	*70%*	*57%*	*46%*	*36%*	*30%*	*64%*	*35%*	*30%*	*38%*	*82%*	*77%*	*67%*	*45%*	*31%*	*27%*	*23%*
MI neg	*0%*	*0%*	*0%*	*0%*	*0%*	*0%*	*0%*	*0%*	*0%*	*0%*	*0%*	*0%*	*0%*	*0%*	*0%*	*5%*	*5%*	*0%*

Figure 5. Clusterings are represented in the columns. A specific cluster is identified by similar numbers on the rows and also by colour coding. The cluster Manual (Low/Med/High) correspond to the cluster of figure 4.

Using both strategic and technical relations is highly useful as a guideline for system design in mechatronic systems. The freedom provided by the use of software implies that more regard can be taken for strategic issues in future systems. The results show that strategic aspects together with ARP can give a relevant clustering of software based products. The exact values in the relations and the weights on strategic versus technical relations however are an open issue for future research.

DSM is very good to visualize the elements and their relations. It is also possible in a qualitative manner to get an idea of what happens when the module boundaries are moved slightly, which may be observed in figure 4. The clustering algorithms using METIS and Matlab are however needed to support in the swapping of rows and columns in an effective manner. The clustering algorithms together with metrics such as ARP give hints to modular solutions, which then may be evaluated qualitatively by the user. Analysis of a set of clusterings with increasing offset makes it possible to order clusters according to module strength. In this process the ARP measure can be used for identifying balanced clusters with a balanced set of elements. Using ordered clusters it becomes possible to assign specific elements to a chosen level of proximity. Two elements can be placed in the same ECU, on the same bus, or separated by a gateway in the network, depending on the strength of the cluster.

An important point is the need to analyze several functions simultaneously. In this paper the relations within a single function have been studied. With several functions using the same elements it is very important to make a common analysis to ensure that sub-optimization is avoided. This analysis will introduce more elements and also extend the number of relations as each function would correspond to a technical DSM. If technical relations are ambiguous or weak, strategic relations will play a more important role.

5 Conclusions

DSM and related measures is a very good tool for analysis of mechatronic systems. The clusters found in this study, with a high ARP score, do all more or less seem sensible and thus could represent a real modular structure of an ACC system in a heavy duty truck.

Earlier studies have suggested modularization based either on the properties of the product or on external requirements. This work shows how the two views can be combined as technical and strategic modularization. In a mechatronic system, including software, the strategic aspects may be stronger module drivers than technical concerns considering that software is more easily decomposed than mechanical components. Recognizing ARP as a useful measure it is shown how both technical and strategic aspects can be considered in a combined DSM matrix to create an ARP-strong clustering. Modularization based on the separate aspects results in solutions with lower ARP values than using a combined DSM as shown in figure 5.

As shown in this paper, mathematical methods and algorithms can be extremely useful to bound the solution space and support further qualitative reasoning. Furthermore, the introduction of an offset in the analysis allows reasoning about the strength of relationships and how to weigh technical and strategic aspects together. For good final results the technical and strategic aspects should be analyzed both individually and in combination and then be manually weaved, enabling cost-efficient solutions for the automotive industry.

References

[1] Pahl G. and Beitz W., "Engineering Design – a systematic approach", Springer-Verlag, 1996.

[2] Ulrich K., "The role of product architecture in the manufacturing firm", Research Policy, vol. 24, pp. 419-440, 1995.

[3] Blackenfelt M., "Managing complexity by product modularisation", PhD Thesis, Royal Institute of Technology (KTH), Stockholm, 2001.

[4] Pimmler T.U. and Eppinger S.D., "Integration analysis of product decompositions", ASME Design Theory and Methodology conference 1994, DE-Vol. 68, 1994.

[5] Erixon G., "Modular Function Deployment – a method for product modularisation", PhD Thesis, Royal Institute of Technology (KTH), Stockholm, 1998.

[6] Newcomb P.J., Bras B. and Rosen D.W., "Implications of modularity on product design for the lifecycle", ASME DETC, DETC96/DTM-1516, Irvine, August 18-22, 1996.

[7] Karypis G., "METIS: Family of Multilevel partitioning algorithm.", Internet reference http://www-users.cs.umn.edu/~karypis/metis/, accessed 2003-01-23.

For more information please contact:

Michael Blackenfelt   Scania, RESC   SE – 151 87 Södertälje   Sweden
Tel: Int +46 8 553 80328   Fax: Int +46 8 553 81395   E-mail: michael.blackenfelt@scania.com

Ola Larses   Scania, RESC   SE – 151 87 Södertälje   Sweden
Tel: Int +46 8 553 89465   Fax: Int +46 8 553 81395   E-mail: ola.larses@scania.com