INTERNATIONAL DESIGN CONFERENCE - DESIGN 2002
Dubrovnik, May 14 - 17, 2002.

DESIGN 2002

A STUDY OF THE EFFECTS OF DIFFERENT TYPES OF SYSTEM ARCHITECTURE ON THE DEVELOPMENT PROCESS

C. Grante and N. Bylund

Keywords: product development, system architecture, concept phase, design theory, organ structure, design structure matrix (DSM)

1. Introduction

Many companies use the same development process in all their departments, although their products do not all share the same system architecture. This often causes problems since the architecture of the product affects the development organisation and the development process [Eppinger and Salminen, 2001]. "Design involves, among other things, people, products, tools and organisations." [Blessing, Chakrabarti and Wallace, 1998]. This study has investigated how product architecture affects the development process and thereby the tools and organisation. Characteristics, similarities and differences between different product and types of system architecture are addressed. Architecture can be integrated, distributed, or partially distributed and partially integrated (mixed). The hypothesis is that types of architecture differ in the interaction (spatial, energy, information and materials) [Eppinger, Whitney, Smith, and Gebala, 1994] among their parts. To verify the hypothesis, studies were made at organ level [Hubka, Andreasen and Eder, 1988]. Studies were conducted at Volvo Car Corporation, since a car contains systems representing all three types of architecture.
The organs are function carriers during the conceptual phase, permitting multiple possibilities of realisation. The studies at organ level give results that are independent of detail design. This gives an understanding of the effects of different architectures in the conceptual design phase.

2. Method

In order to identify the research methods needed and appropriate subjects, exploratory research was used. Methods were selected specifically to investigate the influential and interrelational differences between the different types of architecture: integrated, distributed and mixed.

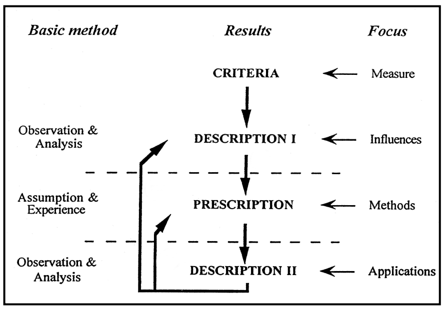

Figure 1. Design research methodology framework [Blessing, Chakrabarti and Wallace, 1998].

2.1 Research method

In our research, we followed Blessing, Chakrabrit and Wallace's approach to engineering design research [Blessing, Chakrabrit and Wallace, 1998] with the four steps: criteria, description I, prescription and description II (see Figure 1). Success is measured in the criteria; in description I the problem is analysed, in prescription a solution is proposed, and in description II the proposed solution is evaluated with respect to initial criteria.

2.2 Design Structure Matrix

This paper presents the analysis (description I), which is an interesting intermediate result. The Design Structure Matrix (DSM) method [Steward, 1981] is used for the study of interaction in each of the three types of architecture at organ level. With the DSM, the interaction between organs can be visualised. Tools exist for manipulating the DSM in order to obtain information about the most appropriate design approach: iterative, parallel, or sequential.

2.3 Organs

The decision to investigate the different types of architecture at organ level was taken because organs represent functional implementation in products, although they are not detail designed, giving an understanding of design in early phases. Organs are often called function carrier and expresses how the system realise its functions [Hubka, Andreasen and Eder, 1988]. Organs are not the same as components since functions are often realised by several components. One component, however, can also contribute to many functions. Sometimes a component is the same as an organ. An example from Mørup [Mørup, 1993] of the organ structure for an electric shaver is shown in Figure 2.

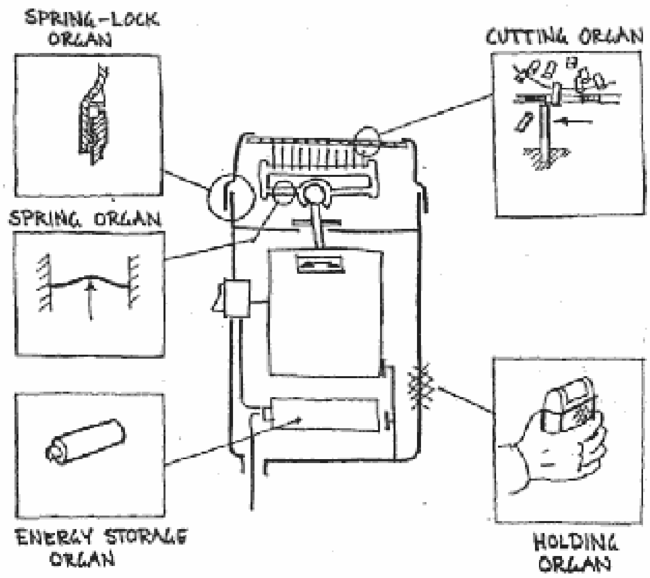

Figure 2. Example of an Organ Structure [Morup, 1993]

2.4 Research objects

Distributed architecture is exemplified by active safety functions in a car, partially distributed and partially integrated (mixed) architecture by the car's chassis functions, and finally, integrated architecture by the car body.

2.4.1 Distributed architecture

The active safety systems that exemplify distributed architecture are ABS, Traction Control System (TRC), Vehicle Stability Control (VSC) and Emergency Brake Assist (EBA). These systems are distributed with functions that are shared between them. The functions are realised by single organs controlled by software and electronics, thereby creating a distributed architecture. The organ structure is straightforward for the sensors and actuators, see Figure 3, but more organs are needed to realise the systems. In active systems, logic and communication are required; these are also considered as being organs. An example of this is that each system has its own logic organ, which at component level can be realised as software or electronics. There is also an organ to determine the communication standard.

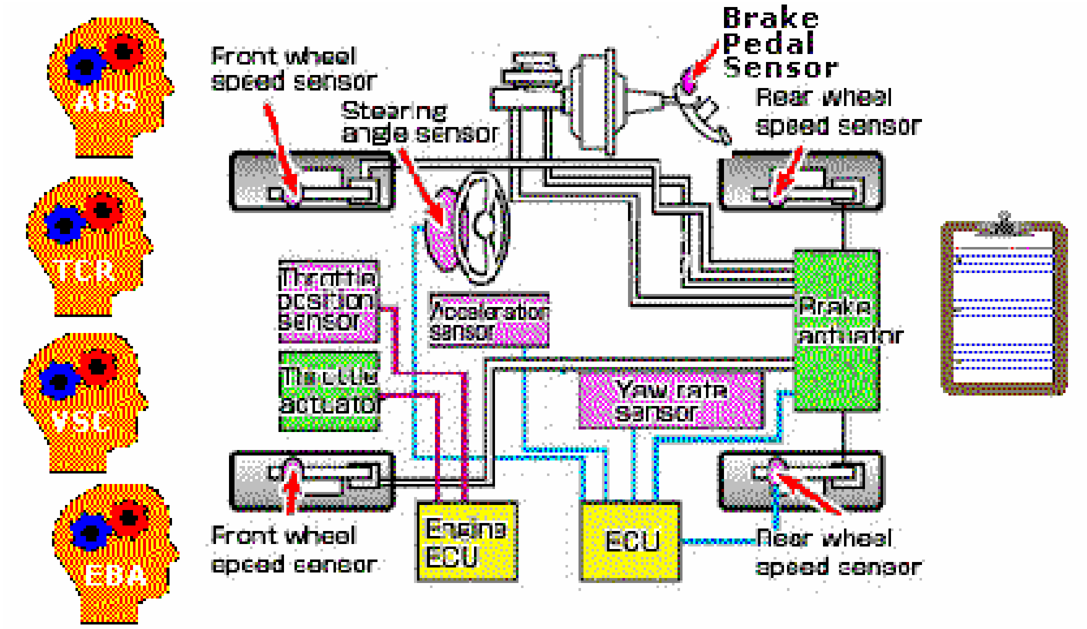

Figure 3. Active safety systems' organs structure. Active safety systems are an example of distributed architecture

2.4.2 Integrated architecture

The system that best exemplifies the integrated architecture is the car body in white. The car body is a self-supporting mechanical structure, which means that every part contributes to the overall performance. If one part is modified, the performance of the structure changes and the other parts will be loaded differently. This makes the car body a suitable example of integrated architecture. The organs chosen are beams, joints and panels [Bylund, 2001]. The joints are defined as the junction off more than one beam, including a portion of the beams themselves. The car body organs, beams, joints and panels interact spatially with each other. Forces are also transferred between these parts, creating deflections and thus a certain compliance (opposite of stiffness), at high force levels, eg, in a crash situation, the organs deform plastically and new spatial interaction occurs and has to be managed appropriately.

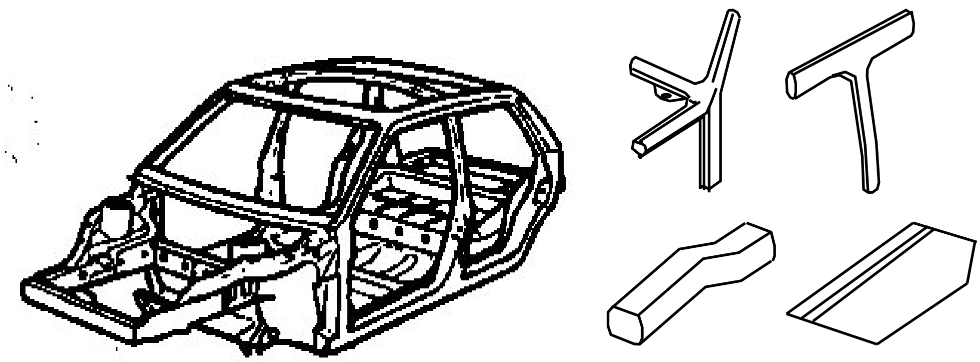

Figure 4. Organ structure of a car uni-body. The car uni-body represents integrated architecture

2.4.3 Partially distributed, partially integrated architecture

The chassis systems investigated are the steering system and the braking system (see Figure 5). The purpose of the steering system is to convert the steering wheel rotary motion into a turning motion of the vehicle's front wheels, and the purpose of the braking system is to reduce the speed of the moving vehicle, as well as keeping it stationary if it has already come to a stop. Both these systems have organs that are of a structural or rather mechanical nature that is reminiscent of integrated architecture as well as information exchange organs similar to those in distributed architecture. This represents mixed architecture, expressing issues not present in the purely distributed or purely integrated types of architectures alone. The organ structure is not straightforward, it is a combination of the distributed architecture's and the integrated architecture's organs.

2.5 Data collection

Three methods of data collection were employed: examining drawings, examining systems from disassembled cars and interviews with experienced designers. The drawings and disassembled systems provided the input for the organ structures and thus the DSMs. The organ structures and DSM couplings were re-evaluated by participating observation [Blessing, 1994] together with Volvo designers.

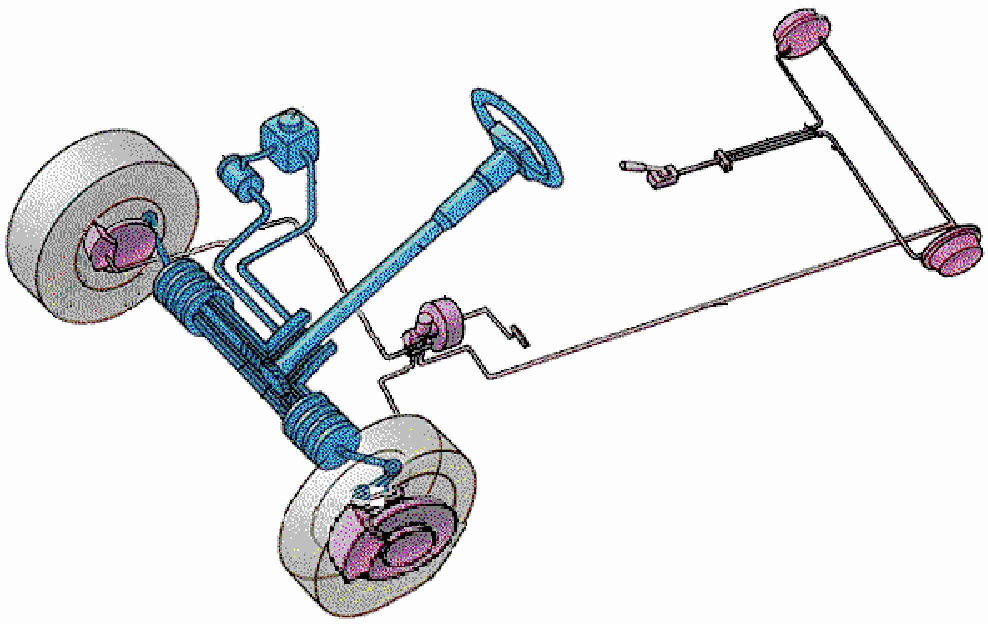

Figure 5. The brake and steering chassis system. The brake and steering system is an example of mixed architecture

4. Results

Parallel, iterative and sequential tasks were identified using the DSM method [Steward, 1981]. Various tools support this. These tools [http://web.mit.edu/dsm/] arrange the order of the organs in the DSM, visualising parallel tasks with black frame blocks, and iterative tasks with grey shaded blocks. Non-overlapping blocks and tasks are sequential (see Figures 6-8).

4.1 Distributed architecture

To design a distributed architecture, firstly an iterative loop is needed to develop the functional logic. This can be seen as the grey box in the upper left corner in Figure 6. Logic is the functional behaviour of the product. A large part of the logic consists of treating cases where the different systems demand different actions from the actuators. Secondly, the interface requirements must be determined. The interface requirements need the logic as input since it decides the type and quantity of information that will be exchanged. The third step, which is the main one and the most resource consuming, is the development of the input and output organs. Fortunately, the input and output organs can be developed in parallel, as shown in Figure 6. Finally the interface layout is established.

	VCS Function	ABS Function	TCS Function	BA Function	Communication Standard	Brake Actuator	Energy	Brake Pedal Position Sensor	Wheel Speed Sensor	Steering Wheel Angle Sensor	Acceleration Sensor	Yaw Rate Sensor	Throttle Position Sensor	Throttle Actuator	Comunication dimensioning
VCS Function	■	1	1	1											
ABS Function	1	■	1	1											
TCS Function	1	1	■	1											
BA Function	1	1	1	■											
Communication Standard	1	1	1	1	■										
Brake Actuator	1	1	1	1	1	■	1								
Energy						1	■								
Brake Pedal Position Sensor		1			1			■							
Wheel Speed Sensor	1	1	1		1				■						
Steering Wheel Angle Sensor	1				1					■					
Acceleration Sensor	1				1						■				
Yaw Rate Sensor	1				1							■			
Throttle Position Sensor	1		1		1								■		
Throttle Actuator	1		1		1		1							■	
Comunication dimensioning	1	1	1	1	1	1	1	1	1	1	1	1	1	1	■

Figure 6. Distributed architecture at organ level, illustrated by automotive active safety systems, ABS, TCS, VSC and BA

4.2 Integrated architecture

All organs interact, so iteration is necessary during the development process. The symmetry of the matrix, see Figure 7, shows that this is, in principle, a stiffness matrix for a mechanical structure representing connections between organs. Figure 7 shows that no individual organ can be tampered with without the global performance of the system being affected.

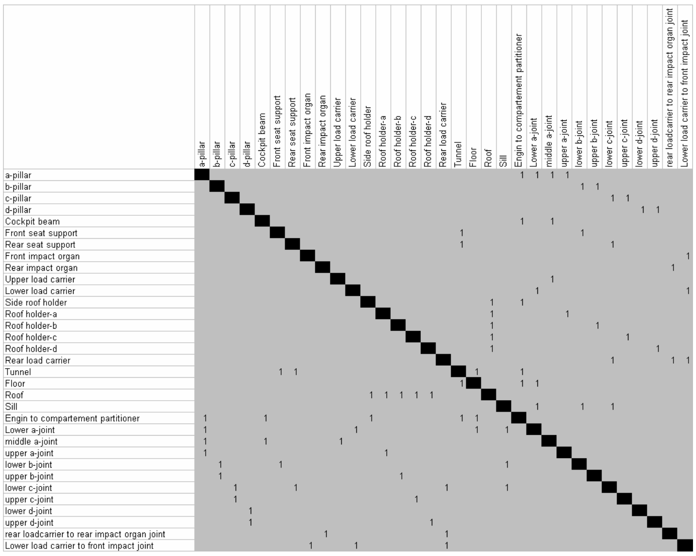

Figure 7. Integrated system at organ level, applied to the car body in white

4.3 Mixed architecture

Each system contains mechanical parts that act as types of integrated sub-architecture; these can be seen as iterative blocks in Figure 8. The systems also exchange information with each other, resulting in couplings between the systems. Due to these couplings, mixed architecture can not be designed in parallel but sequentially instead. There are similarities between the chassis system investigated in this paper and the case study in [Eppinger, Whitney, Smith, and Gebala, 1994].

	Driving cycle	Steering signal	Steering intention	Steering force	Wheel interface	Friction	Heat dissipation	Break actuation force	Brake torque	Chassis interface	Break signal	Break intention
Driving cycle												
Steering signal	1		1	1								
Steering intention		1		1								
Steering force	1	1										
Wheel interface				1		1	1					
Friction							1	1	1	1		
Heat dissipation	1				1	1				1		
Break actuation force									1			
Brake torque	1					1		1				
Chassis interface				1			1		1			
Break signal	1							1				1
Break intention	1										1	

Figure 8. Partially integrated and partially distributed architecture at organ level, applied to chassis systems

5. Discussion

Integrated systems are characterised by their complex and poorly defined interfaces, which couple them together tightly, making their global performances dependent on sensible interactions between their organs, see Figure 7. In our example, beams, joints and panels interact spatially with each other, forces are transferred between them creating deflections and at high force levels, eg, in a crash situation, the organs deform plastically and new spatial interactions are created that must be dealt with appropriately. Replacing one organ with another is therefore a complex task; every interaction must be respected if the overall performance is to be preserved. Changes in a purely integrated system can be seen as manipulating a Rubik's cube. Changing the place of one colored field moves all other fields. Starting from a state where the colors are randomly distributed on the cubs faces reaching the desired state where each faces has one color is long iterative process.

Distributed systems on the other hand, are characterised by their well-defined interfaces, coupling them together in a more predictable way. In our example, active systems with organs such as sensors, actuators and data boxes, interact with well-defined standardised interfaces such as a communication bus with communication protocol and electrical power of 12 volts. Spatial interaction between parts in these systems seldom occurs. Consequently, changes in a distributed system are relatively easy to carry out. As long as the interfaces are maintained, the design freedom is considerable. An example are the electrical devices in a ordinary household, lamps, radios, vacuum cleaners, etceteras, these can easily be moved around to any desired position without the need of iteration.

Many systems are a mix of integrated and distributed architecture, as is the mixed example, the chassis system. As seen in Figure 8 is these system are built from clusters of interacting organs that are coupled by interfaces giving a sequential system development.

The findings indicate a relationship between the interface and type of system architecture (Figure 9). The extremes are integrated architecture without any specified organ interfaces and distributed architecture with fully defined interfaces. Mixed systems architecture can be closely related to integrated and distributed systems architecture as well as those that are somewhere in-between, see Figure 9.

In the participation studies conducted on the three different types of product architectures, similarities have been observed at the start of the organ design. Every type of product architecture had a high level of transparency, considerable design freedom, and low cost bindings at organ level. During the design of the organ structure, different behaviour can be seen in the different types of product architecture (see Figures 6-8). For distributed systems, the requirement breakdown is specifying requirements for each individual organ independently, a process which demands significant resources. This process is seen in the upper left-hand corner of Figure 6 where the functionality of the organ is decided and thereby the requirements. The requirements are used in the parallel development of the input and output organs and are not changed during the design.

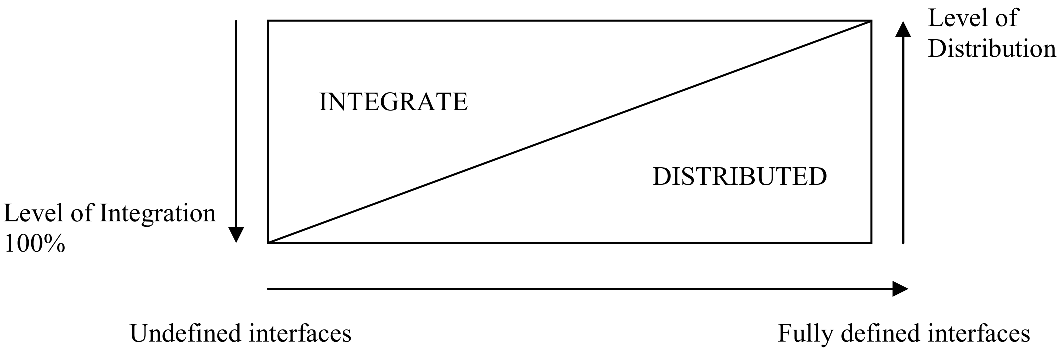

Figure 9. Principal relationship between type of architecture and type of interface

In the integrated system the requirement breakdown is an iterative process, which must go on until the mutual interaction between organs results in the global requirements being met. This means that they are evolving along with the organ design, leading to a significant difference in the architectural design that can be seen in the cost allocation [Andreasen and Hein, 1987]. The cost allocation of integrated systems has a faster ramp-up than the one for distributed architecture. Early cost allocation reduces the dynamic characteristics' flexibility, dynamics, and time on the market. The major reason for this is that the designer has to look further ahead the earlier the cost allocation takes place. The mixed system can be said to have a mixed behaviour. The requirements are iterated in the iteration blocks, with each iteration block delivering requirements for the next block in the sequential design (see Figure 8).

In an integrated system, improved behaviour of one organ can compensate for poor behaviour of another organ, leading to balancing. This demands methods and tools in the early phases that support the requirement evaluation and breakdown. In our example, the interdependent organ requirements have to be balanced until the global mechanical requirements on the car body are fulfilled. This iteration can be achieved in two different ways:

- manual iteration by trying out different performances on the organ and checking the resulting global behaviour until an acceptable balance of requirements is reached.
- optimisation, where an objective function, such as minimum weight in respect of a required performance, is needed as input as well as a concept to start with. System design by distributing requirements to local level using optimisation, also called target cascading [Kim, 2001] has a greater chance of reaching optimum than optimisation carried out at late phases in detail design where only minor changes in thickness or shape are possible.

Iteration also takes place in the development process of distributed architecture, but it differs from the integrated architecture in that the iteration is organ internal with unchanged interfaces. An improvement in one organ in distributed architecture cannot compensate for deterioration in another. The key to distributed design lies in the maintenance of the interfaces and this requires tools.

The mixed systems need both of the above tools and an additional planning tool since the tasks are performed in sequence. The mixed system has an easier iteration and requirement breakdown process but does require resource management. The management of resources when designing mixed architecture is more difficult because there is more sequential development. In practice this means that the different groups working on the product can not do so simultaneously. For bigger projects and organisations, this demands management tools, which are often PERT or Gantt chart based.

Acknowledgements

The gratefully received financial support has been provided by Volvo Car Corporation and the Foundation for Strategic Research through the ENDREA program.

References

Andreasen, M.M. and Hein, L., "Integrated Product Development" IFS (Publications) Ltd/Springer-Verlag UK pp. 15-18, 1987

Blessing, L.T.M .,"A process-based approach to computer supported engineering design" PhD thesis, University of Twente, 1994

Blessing, L., Chakrabarti A. and Wallace K., "Designers – the Key to Successful Development.", Springer-Verlag, London, pp., 56-70, 1998

Bylund, N and Eriksson, M, "Simulation Driven Car Body Development Using Property Based Models" SAE paper 2001-01-3046, in proceedings to IBEC 2001, conference postponed to summer 2002 due to 11 September event

Eppinger, S.D. and Salminen, V., "Patterns of Product Development Interactions", International Conference on Engineering Design, Glasgow, 2001

Eppinger, S.D., Whitney, D.E., Smith, R.P. and Gebala, D.A. "A Model-Based Method for Organizing Tasks in Product Development", Research in Engineering Design. vol. 6, no. 1, pp. 1-13, 1994.

Hubka, V., Andreasen, M.M. and Eder, W.E., "Practical Studies in Systematic Design", p10, section 3.2. Butterworths, London, 1988

Kim, H.M., "Target Cascading in Optimal System Design." PhD thesis, University of Michigan, 2001.

Mørup, M., "Design for Quality" Technical University Denmark, Lyngby, 1993

Steward, D.V., "The Design Structure System: A Method for Managing the Design of Complex Systems", IEEE Transactions on Engineering Management. vol. EM-28, no 3, , 1981, pp. 71-74

Christian Grante MSc
Fluid and Mechanical Engineering Systems
Linköping University of Technology, Linköping, Sweden
and
Research and Development, Volvo Car Corporation, Gothenburg, Sweden
Tel: +46 (0)31 594853
E-mail: cgrante@volvocars.com