# Improving Product Development Projects by Matching Product Architecture and Organization

**Bas Oosterman**

Research School
Systems, Organisation
and Management

SOM

# Improving Product Development Projects by Matching Product Architecture and Organization

Bas Oosterman

Uitgever: Labyrint Publication
P.O.Box 662
2900 AR Capelle a/d Ijssel
The Netherlands
fax +31 (0) 10 2847382

Drukwerk: Print: Offsetdrukkerij Ridderprint B.V., Ridderkerk

Layout binnenwerk: Dick Rutgers BNO, Haren

ISBN: 90-72591-99-2

RIJKSUNIVERSITEIT GRONINGEN

# Improving Product Development Projects by Matching Product Architecture and Organization

**Proefschrift**

ter verkrijging van het doctoraat in de
Bedrijfskunde
aan de Rijksuniversiteit Groningen
op gezag van de
Rector Magnificus, dr. D.F.J. Bosscher,
in het openbaar te verdedigen op
dinsdag 18 december 2001
om 13.15 uur

door

Bas Jeroen Oosterman
geboren op 5 augustus 1972
te Meppel

| | |
|---|---|
| Promotores: | Prof. dr. ir. G.J.C. Gaalman<br>Prof. dr. ir. F.P.J. Kuijpers |
| Beoordelingscommissie: | Prof. dr. ir. L.T.M. Blessing<br>Prof. dr. ir. J.M.L. van Engelen<br>Prof. dr. ir. A.C.J. de Leeuw |

ISBN: 90-72591-99-2

*Je afzonderlijke delen zijn niet zo bijzonder*
*Maar als geheel ben je een wonder*
Salman Rushdy

# Preface

Je afzonderlijke delen zijn niet zo bijzonder, maar als geheel ben je een wonder'. Broadly speaking this means: 'The whole is more than the sum of the parts alone'. This not only relates to the topic of this dissertation, but also refers to the PhD. project itself. One might have the opinion that this book is the main result of my project. For me though it is only a part of it. I consider the experience of exploring the scientific field as being equally important. Especially the sum of the final book and all trial and error during the process make the last 5 years complete and a valuable and unique happening for me.
Unfortunately for the reader, the best memories and stories of the scientific journey cannot be found in this book. In order to provide the reader a glimpse of it, I will take the opportunity to mention some of the many people who played an important role. Without them the dissertation would not have been completed. I like to express my enormous gratitude to them.

First of all I would like to thank Wim Klein Haneveld and Martin Land who were my supervisors during my master thesis in econometrics and inspired me to further explore the world of science.

I was lucky to have Gerard Gaalman and Frans Kuijpers as supervisors. In their own way, they intensely guided me in doing research: theoretically, practically, and personally. I think the match between them made the overall supervision even more powerful with respect to this project. The things I learned will play an important role within my future years. Many thanks for your always direct and personal involvement!
I am very grateful to Jannes Slomp who was greatly involved in starting up this project.
I thank the reading committee for their effort and their useful comments on the draft.
Many people at the faculty helped and inspired me during my research. In particular I express my gratitude to Hen van de Water for his thorough and detailed comments on my paper. I would like to thank Dirk-Pieter for the many fruitful discussions, his broad interest, and last but not least his chewing gum!
I thank Julia Harvey for improving my English.
Dick Rutgers professionally has helped me with the layout of this book. It is impressive how he, under high time-pressure, changed my rough layout into a form that I had (implicitly) in mind.

The shaving company is acknowledged for spending two years there. It was a great time! Cees Wesdorp and Michiel Hillen thank you for allowing me to do the research and for your support. Bart Velthuizen and Ton van Veen helped me from the beginning till the end of my stay. I learned a lot from your 'So What' questions, and your skills presenting findings in a clear fashion. Moreover, you made my stay a pleasant one. Maarten Rijken, Oedilus Bisschop, Sint Baron, Johan Beugels, and Johan van Toor thanks for your time, openness and knowledge. Furthermore, I would like to extent my gratitude to the architecture team: Jan Otter, Bernard

Hekstra, Ton van Veen, Per Ambrosiussen, Anne van de Meer, and Pieter Oosting. We worked together intensely with high effort, but there was never lack of humor. Wilfred Ruijsch and Seakle de Vries thank you for your company and transport every day.

I would like to mention Sonny Kwok. I met Sonny at the shaving company and from that time we got along really well. He arranged that I could apply the analysis within Singapore, and from January 2002 we will be colleagues over there. I especially thank Andre Krebbers, Ferdinand Coehoorn, Gan T.B., Choy K.W. for the invitation and their support during my stay there. I look forward to working with you.

My colleagues at the faculty of Management and Organization made my stay a pleasant one. It was always a pleasure to be a member of the cluster OPS and the cluster Business Development. Linda, Hanny, Nynke, and Henny thank you for the secretary support.
Furthermore, I thank the eight floor, the VF, the aio's, my room mate Henrico, the THW-people (fl. 6,-), the successful SPR-volleyball team (second again...), the 'Thompson-discussion-group', SOM, Bram van Dam, and Nelie Schouten.

I would like to mention my friends and family for my life next to work. My volleyball teammates and the people around (Veracles and Lycurgus) played an intense role. It was a pleasure! Hans Geene, and Gerard Smit thank you for your skillful training and coaching.
Furthermore, I thank all 'Stadspark' soccer players for the fun of just playing soccer.

The friends of econometrics, Jeroen, Gerrit, and obviously the 'Warmoesstraat': Koos, Henriette, Birgitte, Mark, and partners thanks for all great moments and friendship.

Jos, Loes, and Mari-janne. You were always there. I dedicate this book to you...

Bas Oosterman
Groningen, 2001.

BasOosterman@hotmail.com

# Table of Contents

# 1 Introduction to the research problem

## 1.1 Research area and scope

It has been widely observed that the development of new products has become a critical weapon for organizations acting in competitive environments. Changing technology, globalization, and more demanding customers are some of the factors that are forcing firms to be innovative (Wheelwright & Clark 1992). It only takes a glance at the television or newspaper to confirm the fact that firms are trying to attract potential buyers with superior technical performance and fashionable styling.

Variety, change and speed are key issues that companies are nowadays having to deal with (Galbraith 1994, Ulrich & Eppinger 2000). The market requires new products that will have customers checking their bank accounts to see if they can afford the change to a new model. Where product development used to be an occasional and unique activity, competitive firms are nowadays favoring faster innovation (Eppinger et al. 1994, Pine 1996, Meyer et al. 1997).

Not surprisingly, efficient development processes are a requisite for survival. As a consequence, companies are focusing more and more on understanding their development processes and research into successful product development is attracting considerable attention (Cooper 1993, Brown & Eisenhardt 1995). In line with this, this thesis will consider how organizations developing physical products can improve their design processes.

*Design literature in general*

In our context, product development includes the process of transforming a vague idea into a newly functioning physical product that can be launched onto market. As a rule, the members of design[1] project teams do all the work and not surprisingly, they face a considerable variety of complicated issues. Numerous studies have been conducted aiming to find the key elements of successful and effective product development. Looking at the overall picture, the available research includes knowledge from two broad areas: engineering design communities, and organizational/management science communities (Smith & Morrow 1999). The first typically focuses on procedures to technically guide engineers in constructing a physical product. The second considers the question of how people can effectively work together. Both have existed side by side for a long time, but all have greatly contributed to our understanding of product development.

*Concurrent Engineering*

Since the eighties, the performance of design projects has dramatically improved due to the Concurrent Engineering approach (Ettlie 1995). Traditional functional barriers have been broken down and project members have started focusing on concurrent (parallel) execution of all design tasks. The approach emphasizes that decisions made by marketing (to take one area) will affect design, purchasing, or production decisions, and such decisions should not be made in isolation from each other. Accordingly, engineering researchers have designed

[1] In this thesis, design and (product) development are similar terms.

and applied tools such as Design for Assembly (or DFX), Failure Mode and Effect Analysis, and Quality Function Deployment in order to guide project members to integrate the decisions made by various disciplines (Ulrich & Eppinger 2000). Similarly, management researchers have highlighted the role of multi-disciplinary teams in easing the exchange of a great amount and variety of information between project members. An overview of Brown and Eisenhardt (Brown & Eisenhardt 1995) shows that they investigated a myriad of organizational issues such as the role of gatekeepers, group thinking, polarity, team tenure (history of working together), team size and team structure. Concurrent Engineering practices now are widespread and commonly applied within innovative firms.

*The 'underlying design structure'*

In the nineties, a small group of researchers involved in product development (Von Hippel 1990, Krishnan et al. 1997, Smith & Eppinger 1997a, Smith & Morrow 1999) noted that merely addressing integration mechanisms is not a universal solution to improving design processes. This was inspired by the observation that it was large and complicated design problems in particular where there was too much overlap and that integration of activities may overtax coordination mechanisms and seriously hamper project performance. They stressed the importance of understanding and reducing the overall need for coordination during the design process.

As a new perspective, they argued that the way in which a project is decomposed into smaller sub-problems has an enormous effect on the speed and quality of the project's outcome. This is probably best illustrated by the following citation from Von Hippel (Von Hippel 1990), the first to shed some light on this issue within the context of product development.

> *Consider how one might partition the project of designing an airplane. In fact, of course, such a project would be partitioned into thousands of tasks. But, for present purposes let us assume that it will be partitioned into only two tasks, each to be undertaken by a different firm. The two alternative partitionings I propose we compare:*
> *Firm X is responsible for the design of the aircraft body and firm Y is responsible for the design of the engine, and:*
> *Firm X is responsible for designing the front half of the aircraft body and engine, and firm Y is responsible for the back half of each.*
> *Taken together, each of these proposed partitionings has the same project outcome – a complete aircraft design. But the two differ greatly with respect to the interdependence of the two tasks specified. The second alternative would require a much higher level of problem-solving between the two tasks. For example, many design decisions affecting the shape of the "front half" of an aircraft body could not be made without forcing related changes on the designers of the back of the body and vice versa: The two halves cannot be considered independently with respect to aerodynamics. In contrast, the detailed design of a complete aircraft engine is much less dependent on the detailed design of a complete aircraft body. As a direct consequence, I suggest, engineers would think the former partitioning far more efficient than the latter. Indeed, faced with the latter proposed division, experts would be likely to throw up their hands and say, "It can't be done that way".*

Decomposition strongly impacts on how effectively team members are able to combine their knowledge and solutions in respect of joint designs, and has recently become a frequent object of study. The underlying structure of the design problem is generally viewed from two interrelated perspectives: the structure of the product (product architecture), and the structure of the organization (the division of work). However, these perspectives are often considered separately. Engineering science thoroughly investigates issues involving product architecture, and organizational science extensively studies effective organizational structures. Despite their strong interrelationship, communication between both bodies of knowledge is very rare in product development literature. The focus of this study therefore is to link the two streams and come to new and applicable insights enabling understanding and handling of complexity in design. Below we will describe the basics of product architecture and the main principles of organization and then the available research linking the two.

*Product architecture*

Engineering literature contains a tremendous amount of technical knowledge describing the technical structure of, for instance, an airplane. The decision of how a product is to be split up into smaller building blocks (e.g. the body and the engine) is considered an essential one in the technical design process (Suh 1990, Pahl & Beitz 1996). The structure of a product is generally referred to as product architecture (Ulrich 1991, Ulrich 1995). It describes what building blocks make up a product, and specifies how these blocks interact. The architecture of a personal computer, for instance, includes a description of its blocks (monitor, the keyboard, the hard disk etc.) and illustrates how these interact (work together) in order to obtain a correctly functioning computer.

In recent years, product architecture has attracted a lot of attention and many scholars have stressed its high impact on the manufacturing firm. Perhaps the most influential characteristic of architecture is 'the amount of dependence' between the building blocks (Ulrich 1995). At the two extremes, products may be modular or integral. The building blocks of modular products have simple or few interactions and can be easily combined into an overall product. On the other hand, the blocks of integral products have very complex relationships and combining them into the overall product represents a hard task.

Many designers are nowadays striving to develop modular products in response to the complexity of the problem. Modular blocks can be designed, produced, and tested fully in parallel processes and easily assembled into an overall product. When such a structure is found, the business processes becomes less complex since people can focus in complete isolation on a relatively small building block, instead of continuously having to consider the whole product.

However, there are many reasons why products are designed in a more integral way. A focus on low unit costs, low weight and fashionable styling as well as technical limitations may force designers to integrate building blocks in order to establish an optimal overall performance. An instant throwaway camera, for instance, is more integrated than an SLR camera in order to save unit costs and weight. Similarly, a traditional personal computer is more modular than a small and lightweight handheld computer. However, integral architectures have a counter effect: the high complexity involved. Due to the high dependence between the building blocks, people can hardly concentrate on one building block at a time, but have to constantly consider the overall product.

As a direct consequence of the trade-off between modular and integral designs, many products are hybrid (Sosa et al. 2000). These are not fully modular or integral and lie somewhere between the two extremes (Ulrich 1995).

Many engineering studies specify a variety of reasons as to why and how product building blocks interact, and stress the implications of product architecture for the firm (Erixon 1998, Blackenfelt 2000). This knowledge is highly detailed and useful, but generally very technically oriented. Little attention has been paid to the detailed consequences of not fully modular product architectures on organizational aspects.

*Organization of design projects*

Organization scientists focus on the fact that product development is generally not a job for a single person but involves a great number of people. All these people do a part of the job, and preferably something in which they are specialized. One does the mechanics, another the electronics, and somebody else considers the production aspects. It is generally argued that the better the work of these people is coordinated, the better the performance of the overall project will be. The central question of how design projects should be organized in order to facilitate coordination has attracted the attention of many scholars. A major issue in this field is how the overall project can be effectively divided into smaller pieces of work.

Inspired by classic organizational theory (Thompson 1967, Galbraith 1973, Mintzberg 1979) it is generally advised that (large) design projects are decomposed into relatively independent design teams that are able to work in parallel to a high degree. Members of such design teams can work together effectively and have little need for involvement with other teams. Many authors stress that the success of such a strategy depends critically on how well the remaining interdependencies between the design teams are managed (Thompson 1967, Galbraith 1973, McCord & Eppinger 1993, Eppinger et al. 1994). Coordination between the design teams in order to obtain a correctly functioning overall project team is called system-level coordination and will be paid specific attention here.

Recently, scholars have argued that system-level coordination is often poorly understood within design projects. It has been observed that project teams are often formed on the basis of intuition and prior experience, and designers pay little explicit attention to how to solve interactions between the teams (Eppinger et al. 1994, Smith & Eppinger 1997a, Smith & Morrow 1999). This seriously hampers project performance and is recognized as an important variable in achieving major improvements. In line with this, so-called Design Structure Matrix (DSM) studies have been conducted that model system-level interactions between design teams in great detail and propose options for improving the design project. Based on a clear representation of the interactions between the teams, two types of recommendations are made:

- Improve the coordination of existing interactions between design teams.
- Reduce the interactions between the design teams such that less system-level coordination is required and working in parallel is facilitated.

The outcomes of these studies can be used in practice as lessons learned and seem to be very effective for a company in achieving improvement of future design projects.

These studies and recommendations are very promising. Remarkably though, they pay very little detailed attention to the underlying architecture of the product that has been designed (Oosterman et al. 1999, 2000, Sosa et al. 2000).

*Linking product architecture and Organization: the gap*
Having discussed both streams separately, the available research that links the two will now be illustrated. Product architecture and the structure of a project team are strongly interrelated. This becomes evident if we realize that most design tasks refer to the design of a piece of the product (Galbraith 1973). However, remarkably little research has been conducted into this relationship.

Management studies report that effective firms should match the structure of the organization to the structure of the product (Galbraith 1973, Henderson & Clark 1990, Gulati & Eppinger 1996). The previous example of the airplane for instance, indicates that 'intelligent' firms should group the work around relatively independent product blocks. In addition, Novak and Eppinger (Novak & Eppinger 1998) have shown that within the automobile industry, highly successful companies clearly mirror the organizational structure in the product structure.

A number of studies also recognize that interactions between building blocks create a need for system-level coordination between design teams. Scholars in product development stress that modular products permit the parallel functioning of design teams, which is hardly allowed for in integral product architectures (Wheelwright & Clark 1992, Ulrich 1995, Gulati & Eppinger 1996, Sanchez 1999a). Moreover, a well-known study by Henderson and Clark (Henderson & Clark 1990) showed the crucial importance of design teams having effective communication channels for managing the technical interactions between the product blocks. The effectiveness of these channels was the most important variable in explaining the success (or lack of it) of innovative projects within the photolithography alignment equipment industry.

To sum up, the above studies all underscore the managerial importance of architecture and its consequences for system-level coordination between design teams. The lessons are obvious, but all are at a relatively high level of abstraction. For actual analysis and improvement of design projects, these lessons lack sufficient levels of detail. For instance, it would be very helpful to understand how a particular company should organize and coordinate their design project for a given (non-modular) product architecture, or to understand how they may have to change their product architecture in order to obtain organizational benefits. Yet, despite what seems an obvious assumption, this understanding is not readily available in the literature (Pimmler & Eppinger 1994, Erixon 1998, Sosa et al. 2000).

There is useful detailed knowledge in engineering and organizational literature, but in both cases it is one-sided. Engineering design methodologies describe product architecture in great technical detail, yet do not explicitly link these to the consequences for coordination during a design process. Alternatively, organizationally oriented studies clearly represent and address the issue of system-level coordination, but do not reflect the underlying product architecture.

Taken together, what is needed is a subtle linkage between engineering science and organizational research. The challenge of how to abstract, combine, and apply these two bodies of knowledge still remains.

## 1.2 Research questions and objectives

The aim of this research is to provide a theoretical and practical means of improving the design processes of complex physical products. For these products, project teams are usually split up into a number of smaller design teams that each do a part of the job. We shall focus on gaining an understanding of the system-level coordination between the design teams during the design process. This study's contribution will be to add the aspect of product architecture to the issue of system-level coordination.

The available research in this area shows that system-level coordination is often poorly understood within design projects, but is an important variable in increasing speed and quality. From the literature we know that architecture and organization are highly related and furthermore that a clear match between product architecture and system-level coordination is of crucial importance for good project performance. However, these studies lack sufficient detail to be of use in understanding and improving actual design processes.

There are thus two important issues to address. The first is the question of how product architecture can explain system-level coordination during a design process at the level of subtle detail. The second is supporting the project team or management in generating options that improve system-level coordination between the design teams. The aim of this thesis is to improve the design process by means of a thorough understanding of the underlying architecture and its implications for the organization. The main assumption is that system-level coordination activities can be better adapted towards the architecture of a product, and that product architecture can be altered to ease system-level coordination.

As with the previously discussed DSM models, the assumption is that answers to the above issues can be obtained by a detailed and useful representation of product architecture. The following research question has thus been formulated.

> *How can the particular architecture of a product be represented such that it offers a clear understanding of the required characteristics for system-level coordination during the design process, and such that it provides a vehicle for generating options to improve the performance of future design projects.*

This research question will be addressed in greater detail in Chapter 4. To obtain an answer to this question, we need to explore the fields of engineering design, organization, and combine the two bodies of knowledge in creating a new approach. The approach then has to be illustrated and explored within an actual design process.
The following research objectives have thus been formulated:

- To define the concept of product architecture and underlying decisions, and explore how the architecture of a particular product can be clearly represented.
- To examine how management science represents system-level coordination within design processes, and which classic organizational principles are available to improve system-level coordination.
- To express what is needed to link the architectural and organizational perspectives.
- To develop an approach that is theoretically adequate for analyzing and improving system-level coordination from the point of view of the underlying product architecture.
- To apply the approach in practice in order to explore how architecture can explain real-life system-level coordination, how architecture is to be used to generate options for improving future design processes, and to explore how these options actually result in better performance (or how this may be measured in the future).

## 1.3 Research method

The above question and objectives indicate that our research has an explorative character. This is not surprising given the limited research available about this topic at the present moment. This will thus have two foci: the construction of a theory and testing it in practice. The thesis starts with a thorough exploration and analysis of the available literature. A case study of a design project for an electric shaver will then illustrate and explore the theoretical findings.

The strategy of doing case studies is particularly suitable for studying specific phenomena in their real-life context and makes it possible to understand complex matter in a detailed way. Depending on the purpose of the research, a single or multiple case study design is possible. A single case study approach has been adopted for this research for the following two reasons. The most important is that, according to Yin (Yin 1994), a single case study is able to confirm, challenge or extend theory on condition that the theory has specified a clear set of propositions as well as circumstances in which the propositions are believed to be true. This is exactly what this thesis aims to do.

Second, there is a more practical reason for choosing one case. It is actually very time-consuming to develop sufficient theoretical understanding, to find an appropriate case and to gain access to it, and to perform the case study with a sufficiently high level of detail. Given the long but finite timeframe available for a dissertation project, a single detailed case study was the only viable option for this thesis's purpose.

Conducting case study research involves considerable preparation and a careful approach. In particular, a clear research design is needed that links the data to be collected to the initial research question.

Critics of case study research believe that single case studies can offer no grounds for achieving reliability or generality in respect of the findings. Others feel that case studies bias findings or are only useful as an explorative tool. However, other scholars have stressed the enormous potential of case studies and introduced procedures that offer the opportunity to establish 'good' research (Eisenhardt 1989, Yin 1994) In general terms, if the case design is sound, and the researcher is explicit about the phenomenon and context of the study, the results of case study research cannot be dismissed out of hand.

Perhaps most important is the role of theory. Many researchers emphasize that the importance of theory cannot be overstated. Proper use of theory will provide a better focus for data collection and give weight to the data. Moreover, if the investigation has a strong theoretical foundation, there is more latitude for the validity of the results across the context in which the theory has been tested.

In addition to a sound theoretical basis, a good case study protocol is essential. Yin (1994) stresses the importance of the case under study matching the conditions of the theory and he proposed a number of rules and data-collection tactics to increase the quality of the research. In line with this, he defined the following criteria for judging the quality of research designs:

- *Construct validity*: establishing good operational measures for the concepts being studied.
- *Internal validity*: establish causal relationship.
- *External validity*: establish the domain to which a study's findings can be generalized.
- *Reliability*: demonstrate that the operations under examination – such as data collection – can be repeated with the same results.

In this research, considerable emphasis has been placed on the theoretical foundation, Yin's principles being applied where possible, and the quality judged on the basis of the above criteria. This will be thoroughly discussed and implemented in Chapter 6. A short overview of the case and the basic steps undertaken will now follow.

The case study was conducted within a well-established company that designs and produces electric shavers. As a consequence of current technological and market trends, its design teams are under pressure to develop better and more complex products, and to develop them faster. The object of our study was an almost finished design project involving a highly innovative waterproof shaver. The project team was large and comprised a number of smaller design teams that each designed one building block of the shaver. The shaver was known to be complex and non-modular. Very broadly speaking, the following steps were performed:

- Documentation of the architecture of the shaver.
- Investigation of system-level coordination efforts (retrospectively).
- Linking of the architecture with system-level coordination activities and generation of options for improvement.
- Exploration of implementation of the options for improvement, focusing on how the effects on increased performance can be measured (in the long term).

Chapter 6 will also describe these steps in much greater detail, and it will be argued that the overall research setting has been designed to ensure that the results are of a good quality.

What needs to be noted here is that the results of the single case study can only be generalized to a limited extent. In fact, to generalize the findings and abstract from the specific conditions of the shaver case a multiple case study design is required. However, what can be done is apply the theories used in a broad way. The logic of the theories permits generalization of the findings taken from one case to other situations (analytical generalization Yin 1994). However, it should be noted that the findings can only be validated by further (multi-case) research.

## 1.4 Outline of the thesis

This thesis is organized around the objectives introduced in this chapter. Broadly speaking, each chapter takes one objective into account. The main structure is depicted in Figure 1.1.

Chapter 2 reviews and analyzes current literature within engineering science. Product architecture is defined and its underlying technical constructs discussed in detail. Furthermore, attention is paid to how to clearly describe the architecture of a product. This chapter will be often referred to in the following chapters.

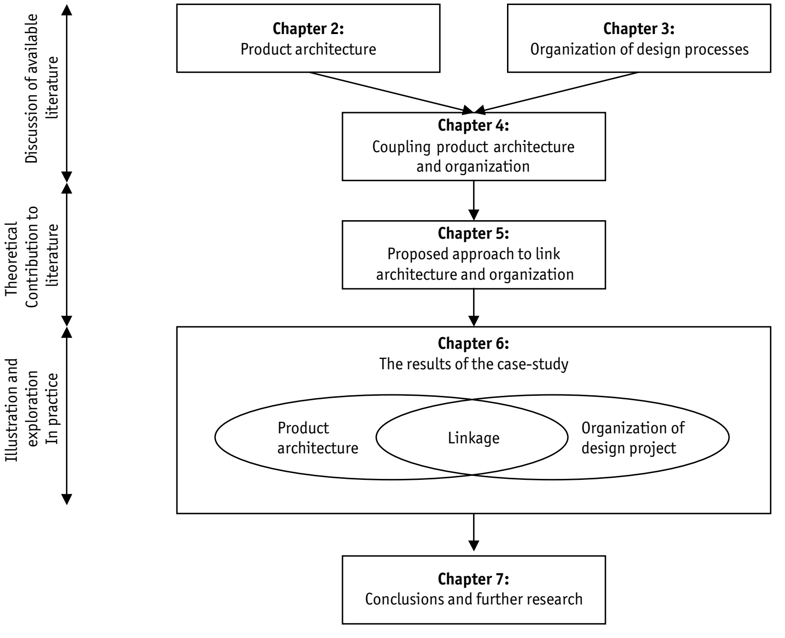
Chapter 3 investigates available knowledge within organization/management science. First, discuss classic organization theory will be discussed, including useful lessons for effective organization and coordination mechanisms. Second, literature describing the coordination problem within design projects will be discussed and options for solving this problem generated.

Chapter 4 describes research that links product architecture and organization. This will be extensively discussed and, based on the previous chapters, what is needed to further explore this relationship will be formulated. The research questions will then be examined again, this time in detail.

Chapter 5 proposes a way to represent and interpret a product architecture that enables the analysis of system-level coordination during a design process. The details provided by Chapter 2 and Chapter 3 will generate the material in this chapter.

Chapter 6 describes the case study. How it has been set up will be described in great detail. The case study results will then be presented, and the results discussed.

Chapter 7 concludes the research. Conclusions are formulated and directions for further research are suggested.




▸ **Figure 1.1** Outline of this thesis

# 2 Product Architecture: Key Concepts and Implications

This chapter examines the first body of knowledge that is needed to explore the relationship between product architecture and organization. We will focus extensively on engineering design knowledge in order to gain understanding about product architecture and to explore how architecture can be clearly represented. In particular, there will be a focus on the precise definition and characteristics of the decisions that determine the architecture of a product. By elaborating these in great detail, a firm foundation will be laid for the later chapters of the thesis, where they will be placed in an organizational context.

To that end, several engineering design methodologies that have been proposed in the literature to guide engineers during the design process will be examined. These models contain extensive technical knowledge on how to construct a physical product and more or less explicitly contain very useful information about product architecture. However, a glance through the existing literature will be sufficient to show that there is no universally applicable engineering design approach. Instead, a great variety of distinct models with different terminology exists, making simple accumulation of knowledge a hard task. In this chapter a selection of engineering design methodologies will be made and the essential elements carefully described in order to enable valid interpretation of the constructs. Several methodologies will be discussed and compared in order to be able to choose a useful set of definitions. The following steps will be performed:

- An outline of problem solving will be presented in order to understand the general principles of engineering design methodologies.
- The key concepts of two distinct well-known design methodologies – VDI Design, and Axiomatic Design – will be provided, discussed and compared.
- Ulrich's (1995) widely accepted definitions of product architecture will be described, and based the previous discussion of the engineering methodologies the definitions for this study will be chosen. Furthermore, how a particular architecture can be represented will be explored and discussed.
- The impact of product architecture on the manufacturing firm will be summarized in order to indicate what architectural decisions are broadly contingent upon.
- The various steps will be summed up.

## 2.1 Problem solving in general

Researchers in product development (and in particular in engineering science) conceptualize the design of a new product as a process in which the organization creates and defines problems and then tries to solve them (Alexander 1964, Simon 1981 Steward 1981, Nonaka 1994, Pahl & Beitz 1996, Thomke 1997, Smith & Eppinger 1997a)[2]. Despite each problem-

[2] Note that publications of Simon related to this topic trace their roots in the early sixties and were one of the first with this point of view. However in this thesis is referred to 'the Science of the Artificial', published in 1981.

solving process being unique, it is commonly believed that all problem-solving processes share common characteristics. This section describes some basic principles of problem solving. This will be helpful for an understanding of the more specific engineering design methodologies, and will also be useful for the organizational theories in Chapter 3. Presently the following issues will be considered:

- Introduction to problem solving.
- Hierarchies in problem solving.
- Link with product development.

### 2.1.1 Problem-solving basics

Problem solvers are concerned with how things ought to be (Simon 1981). They try to realize desired situations. Take, for instance, the case of somebody being in Drachten but aiming to be in the city of Groningen. He or she takes action and catches a bus that brings him or her to the destination station. In problem-solving terminology, catching the bus is a description of a process that takes you from one state (being in Drachten) to another state (being in Groningen). Accordingly, a description of a future desired state will related to how the problem is presented (Simon 1981). In general the problem-solving task is to discover a sequence of processes that will realize the goal starting from the initial state. In order to find a solution, people generate alternative actions and test these against a whole range of criteria. For instance, when you have decided that you want to be in Groningen, there are many alternatives for taking you there. What happens in practice is that one generates alternatives (e.g. going by plane, car, train, bus, bike or on foot.) and these are subsequently tested against a whole range of criteria (e.g. total costs, traveling time, schedule, or availability). Finally, an appropriate alternative is selected and you implement your decision. The process of generating alternative solutions and subjecting them to a general test is referred to as generate-test cycles (Simon). The literature dealing with engineering design shows many small variations on this theme (Pahl & Beitz 1996, Blessing 1996) but broadly speaking they all agree upon the cyclic character of finding a solution. The concept of cycles will return later in this thesis (section 2.2, and Chapter 5).

According to Simon, problem solving is a matter of trial and error. People build up associations between particular states and specify actions that have to realize these changes. In general, the path towards finding a solution is not chosen blindly, but largely shaped by experience or rules to do with which actions should be tried first. As Simon points out:

> *All that we have learned is that human problem solving involves nothing more than varying mixtures of trial and error and selectivity.*

In line with this, many researchers have examined the general rules of effective problem solving. These aim to propose methods that are helpful for finding a solution for a particular problem in an efficient manner. There is obviously a wide range and variety of approaches, though a concept that frequently appears in the context of effective problem solving is that of hierarchy (or decomposition). This will be discussed below.

### 2.1.2 Hierarchies

This section describes two hierarchical concepts that are suitable for complex problems. Each concept emphasizes different aspects and is often concurrently applied. The two are useful for the understanding of product development and in particular for product architecture. The following will be described:

- The vertically arranged layers involved in decision-making.
- The horizontally arranged sub-systems of Simon's hierarchical concept.

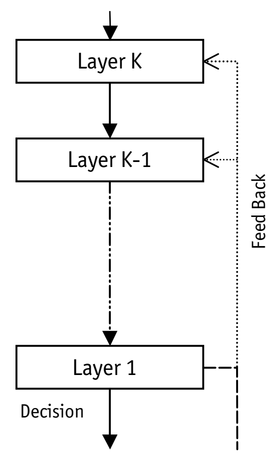
*Layers*

Mesarovic (Mesarovic et al. 1970) states that in real problem-solving situations, people generally do not know the exact consequences of alternative actions. However, postponement of the decision as to what action to implement simply implies that no action is preferable. He illustrates the resulting fundamental problem-solving dilemma below:

> *On the one hand, there is a need to act without delay, while on the other, there is an equally great need to understand the situation better.*

This dilemma can be solved if the solution to the original complex problem is substituted by the solutions for hierarchically arranged simpler sub-problems. That is to say, one defines a sequence of sub-problems for which the solution of a particular sub-problem completes the specification of the subsequent sub-problem that in turn can be attempted. The original goal is achieved when all sub-problems are solved. For instance, in order to reach Groningen you may first make the decision to you go by bus, and then decide where and when you will start the journey (depending on the available buses). These decisions together determine how you will get to Groningen: the solution of the initial problem.

Mesarovic refers to such a hierarchy of decisions as a hierarchy of decision layers. Figure 2.1 shows a decision problem that is partitioned into k layers of sub-problems, where the output decision problem k is needed to specify decision problem k-1. Feedback between the decision units is plotted as dashed lines.




▸ **Figure 2.1** Hierarchical layers according to Mesarovic

Decisions at different layers generally have relatively different characteristics. Compared to a lower layer, a higher layer of decision making:

- is concerned with a larger portion or broader aspects of the overall problem.
- has a longer decision period.
- is concerned with slower aspects.
- is less structured, has more uncertainties, and is more difficult to formalize quantitatively.

It may be relevant to mention that the term sequence of sub-problems must not be confused with a necessarily sequential, top-down way of problem solving. A problem may be attempted by moving up but also by moving down the layers. Moving down the hierarchy is required since a downstream sub-problem cannot be exactly formulated without a solution being found for the upstream problems. On the other hand, moving from lower to higher layers is necessary since the solution of all sub-problems implies a solution being found for the overall problem. The success of a higher layer depends on the performance of the lower layers. If no solution is reached for a specific sub-problem this has to be fed back to a preceding layer or passed forward to a lower layer where it can be fed back to higher layers if necessary. While it is difficult to make firm statements about the decision-making sequence, this still illustrates the need for a multi-layer structure when dealing with complex problems.

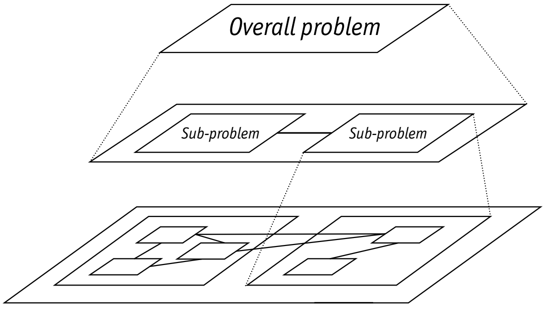
*Simon's hierarchy*

In addition to a hierarchy of layers that focuses on the vertical arrangement of problems, a problem may also be decomposed in a horizontal fashion. Let us return to the traveler. Suppose he has made up his mind and now decides he wants to visit Orchard Road in Singapore. This is a somewhat more difficult problem since there is an enormous range of possible actions that will take him to Singapore. To simplify the problem, he decomposes it into two smaller sub-problems: Groningen-Amsterdam and Amsterdam-Singapore. Each sub-problem can then be solved in relative independence of the other. However, the two sub-problems have to jointly generate a solution for the original problem and therefore have to have a degree of interaction. In the example, the arrival time in the first solution must be established before the departure time in the second sub-problem can be arranged. Furthermore, it is possible that both sub-solutions may together not exceed a certain budget or total travelling time. The decomposition decision is not unique. There may be many alternatives, and these may also lead to alternative processes and alternative solutions. An important feature, however, is the relative independence between the sub-systems. This brings us to the hierarchical concept postulated by Simon (1981).

A system (i.e. a problem) can be viewed as a unit as opposed to its environment. In real life a system is 'open' and has a relation with its environment, specified as system inputs and system outputs. A system can be described as a so-called black box. If that is the case, only its input and output are specified without considering its inner environment. In order to increase understanding of the details, a black box can be split up into smaller black boxes, each with inputs and outputs. In turn, these boxes can be decomposed again until the lowest hierarchical level is reached. This is depicted in Figure 2.2. As a result, a hierarchy is created consisting of an arrangement of interacting sub systems. It should be noted that Simon's hierarchy not only states that a sub-system is part of a larger system (a volleyball player is part of a team, and the team is part of a whole club), but especially considers the interactions between the sub-systems (i.e. how the players interact).

According to Simon, a system is complex when it is made up of a large number of elements that have many interactions. Simon argues further that most systems are to a large extent decomposable. This means that a system can be decomposed into relatively independent sub-systems. Nearly decomposable systems can be split up into sub-systems such that the interactions within the sub-systems are much stronger than the interactions between the sub-systems. Hence an effective approach for dealing with complex problems is to decompose them into smaller subsystems that are each easier to handle, and are relatively independent.

These principles also apply to product development (Von Hippel 1990, Eppinger et al. 1994). Once a problem is decomposed, the interactions between the sub-problems should be well understood and managed. Furthermore, the amount of interaction influences the speed at which the overall problem can be solved. In fact, the number of interactions between the sub-systems determines to what extent the sub-problems can be solved concurrently and therefore strongly impacts on the speed of solving the original problem.




▸ **Figure 2.2** Simon's hierarchical concept

### 2.1.3 Link with product development

The above description of decomposition strategies is very summary but sufficient to provide a panoramic view of the most important concepts informing this thesis. General problem solving literature will therefore not be analyzed further, and instead a closer look will be taken at engineering design methodologies where the basics again occur but now in a more specific form.

- Before doing so, an indication will be given of where the main lines occur later.
- Engineering methodologies distinguish the functions of a product from physical solutions that have to achieve the functions. This is similar to the distinction between a desired state or goal and the process that takes you to the desired state.
- In order to find a solution relating to a product's function, engineering design methodologies prescribe conducting generate-test cycles.
- Engineering methodologies prescribe a whole sequence of stages and steps that need to be taken to design a product. This is fully in line with the layer structure of Mesarovic.
- Within engineering models, problems are constantly being decomposed in smaller interacting sub-problems. Simon's hierarchy applies in particular to product architecture, which deals with how a product can be decomposed into physical building blocks (physically distinct units or chunks of a product) and how these blocks interact.

These aspects will be explained in the following sections.

## 2.2 Design methods

This section will explore a large set of different engineering design methodologies. The overall model area will first be described and the prescriptive design literature then examined. The design methodology of Pahl and Beitz (Pahl & Beitz 1996) will be looked at, as well as the Axiomatic design approach taken by Suh (1990). Finally, the two approaches will be compared and discussed, especially in relation to the definitions and interpretation of the technical constructs that are used.

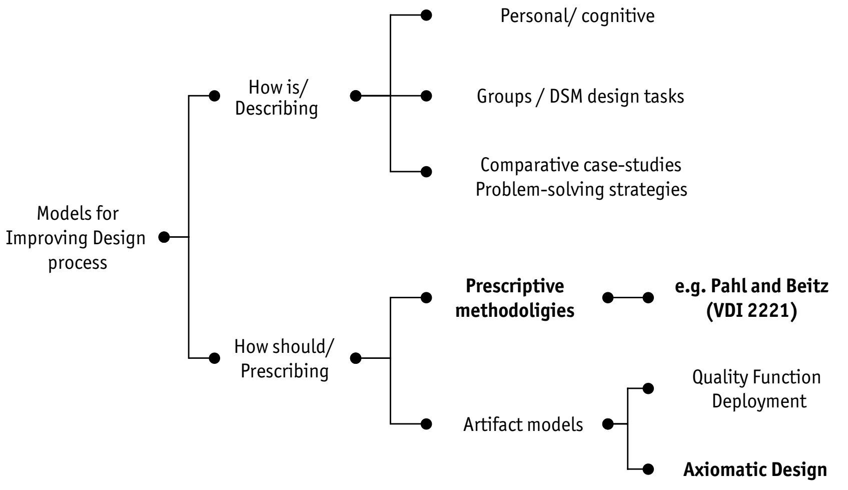
### *2.2.1 Introduction to engineering design models*

A large number of models and methodologies have been developed within the field of engineering science. These ultimately aim to direct decisions and activities during the design process in order to improve performance. To some extent they all include the basic characteristics of problem solving described in the previous section. Not surprisingly though, engineering science offers a great variety of different solutions (Malmqvist 1995, Malmqvist et al. 1996). In general these are divided into two broad categories: describing and prescribing methods (Blessing 1996, Erens 1996, Stake 1999).

Describing models addresses how design processes actually take place. This category can be further divided into cognitive and other studies at an individual level and studies at a group level. The latter may consist of models that illustrate the problem-solving structure of design teams (see the Design Structure Matrix models in Chapter 3.2) or include more general comparative case studies of problem-solving strategies and related performance.

Prescribing methods consist of a collection of formulas that prescribe a rational and structured way of working that purport to lead to an effective and efficient design process. These methods are generally based on the personal experiences of the authors. This category consists of two streams. In the first place, prescriptive methodologies that prescribe a particular process (course of action) that is necessary to bring a product to its final shape. Secondly, artifact models that focus on the outcomes of the process. These describe the evolving states of the product. Quality Function Deployment (QFD) and Axiomatic design are well-known methods within this context. QFD describes a product from the client, design, component, and manufacturing perspectives, and links these points of view (Hauser & Clausing 1988). Axiomatic design addresses particular states of the product by means of modeling and analyzing the evolving decompositions of the original problem (Suh 1990).

Figure 2.3 shows the division of models described above inspired by existing categorizations (Blessing 1996, Erens 1996, Stake 1999). The classification must not be seen as exclusive, but rather as highlighting some relative differences. Putting prescriptive literature and descriptive literature into opposing camps would be too rigid an approach. There are some good reasons why these have at least some overlap. In fact, the prescriptive models originate from the personal experience of designers in the field and thus are implicitly based on practice. Furthermore, it is reasonable to expect that the teaching of prescribing models to engineers will have affected their way of working. Moreover, scientifically speaking, one would expect that both categories to have become highly interwoven over the years. Scholars should explicitly test and adjust prescribing models based on descriptive models and the other way around. However, the latter argument can only have limited validity since prescriptive models have rarely taken descriptive models into account (Blessing 1996).




▸ **Figure 2.3** An overview of engineering design models

In any case, it is reasonable to assume that the basic elements of prescriptive literature are valid and can be applied in practical situations. This is important for the purposes of this thesis since the prescriptive models will be pursued here with the aim of the insights being applicable in practice.

The rationale for focusing on prescriptive models here is that these include a precise definition of technical constructs (functions and technical solutions) that are needed to define product architecture in the next section. However, there is more at stake than definition alone. The aim is also to explore the ways that technical constructs such as functions and technical solutions affect the course of the design process of a product according to the prescriptive models. This insight is helpful in enabling a proper interpretation of the characteristics of a product architecture and placing them in an organizational context later in this thesis.

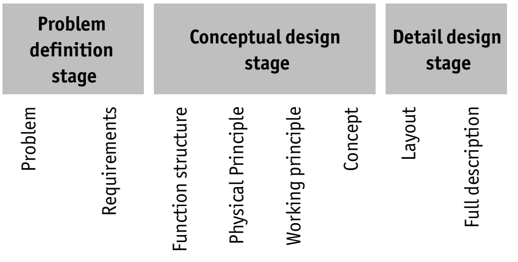
As indicated previously, there is no universally valid design methodology and accordingly no clear uniform definition of the technical terminology. Both prescriptive methodologies and axiomatic design contain very useful insights for defining and interpreting product architecture.

Below the two will be described separately such that their underlying paradigms are made quite clear. The two models will then be compared and described and the foundation laid for the research terminology. The prescriptive methodologies will be described first, followed by axiomatic design.

### *2.2.2 Prescriptive design models*

Prescriptive design methodologies are usually very detailed handbooks describing many stages, activities, and examples, and containing a large amount of technical knowledge. In general, prescriptive design methods divide design processes into stages. Each stage includes the process that takes place between two states of a product. Performing the processes belonging to all stages brings one from the initial state (idea) to the final state (full specification). Furthermore, prescriptive methodologies actively guide human problem-solving activities such as identifying the problem, generating solution alternatives, selection of the best one, and implementation.

Again, there is a great variety of methodologies (Tate & Nordlund 1995, Andreasen et al. 1996, Erens 1996), each with different steps, definitions, foci and different optimal paths for problem solving. Basically, however, the prescriptive methodologies have many common features. Based on an extensive literature study, Blessing (Blessing 1996) concluded that all methodologies roughly fit within a division of three stages: a problem definition stage, a conceptual design stage, and a detail design stage. Each of these stages addresses the process by which a particular state of the evolving product is reached as shown in Figure 2.4.



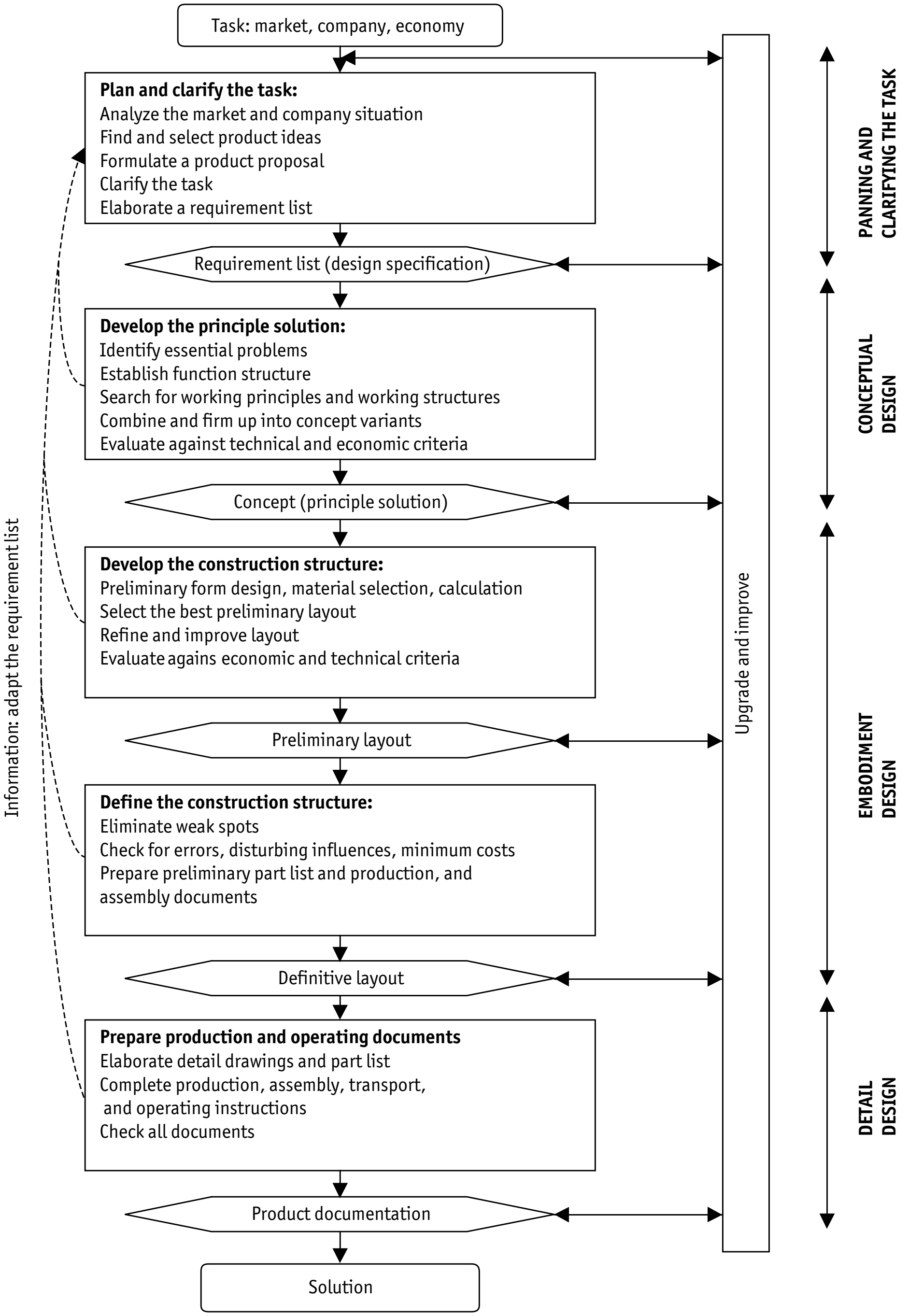

▸ **Figure 2.4** Three stages common to prescriptive models (adapted from Blessing )

All methodologies draw a distinction between the function of a product (what it has to do) and the physical solution (how it is achieved). They usually translate customer requirements into functions, and try to find technical solutions that fulfill these functions such that in the end a fully specified product results.

In order to further elaborate these issues, the VDI design will be focused on, since this is generally representative of a large body of engineering models. VDI was described by Pahl and Beitz (Pahl & Beitz 1996), and is well known in many engineering areas. In fact, VDI is commonly known as 'Pahl and Beitz'.

*VDI Design*

The Society of German Engineers (VDI) devised a methodology (VDI 2221) described in great detail in "Engineering Design: a systematic approach" by Pahl and Beitz. The authors describe four stages including a number of steps guiding the design of a product from scratch to full specification, as illustrated in Figure 2.5. The stages are planning and clarification of the task, conceptual design, embodiment design, and detail design. These will each be described under the headings below. There will be a particular focus on the conceptual design phase where the concepts of functions and working principles are defined in detail.




▸ **Figure 2.5** Steps in a design process according to Pahl and Beitz

*Planning and clarification of the task*

This stage starts with an incentive: bringing a new product to the market that has to be attractive in terms of market conditions and company strategy, and concludes with a list of requirements that need to be fulfilled by the new product. To that end the company will conduct an extensive analysis of the marketplace and situation of the firm and subsequently set a process in motion where product ideas are generated and selected. The most promising idea is then refined by formulating a product proposal that clarifies the product's task. Finally, the company creates a list of product requirements that in turn is used to set the next stage of the design in motion.

*Conceptual design*

The conceptual design stage produces the principle solution required to establish the product requirements. The stage begins with an analysis of the main problem that needs to be solved to satisfy the list of requirements created in the previous stage. The following steps are then taken:

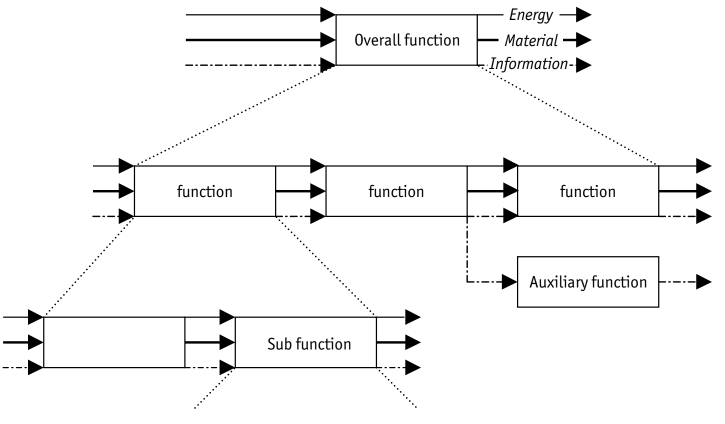
- Construction of the function structure.
- Searching for and selecting working principles.
- Combining the principles into a working structure.

These steps determine the main product structure and will be paid special attention. A designer first needs to formulate an overall product function. A function describes the relationship between inputs and outputs within a system. These inputs and outputs can be categorized into three types: flows of energy, flows of material, and flows of signals (information). A function thus expresses a transformation of energy, material, or information. Functions are preferably described as a verb-noun pair without a preconceived idea of the solution. For instance, 'decrease temperature' may be a function of a refrigerator. This description does not include any indication of how a solution to lowering the temperature can be found.

When the overall function is clearly specified, the design evolves to the stage of decomposing the overall function into smaller functions. These functions are again transformations of energy, material, and information, but at a lower level of complexity. The resulting set of functions can be arranged in a function structure, such as is shown in Figure 2.6. The structure indicates that all functions are part of the overall function and can be connected to each other. The output of the one function becomes the input of the other function. All of the connected flows together constitute the input and output of the overall function.

In addition, Pahl and Beitz classify functions as being main or auxiliary (as can be seen in the function structure depicted in Figure 2.6). They state that the main functions contribute to the overall function directly, whereas auxiliary functions have a more supportive character and contribute to it indirectly. For example 'decrease temperature' is probably a main function of a refrigerator, but may be considered as auxiliary for a personal computer. Hence a computer is not designed to decrease temperature, but its temperature must be lowered to prevent the processor from overheating. The distinction between the main and auxiliary functions affects the prescribed sequence of problem solving. It is advisable to start with the design of the flows of the main functions and afterwards address the auxiliary flows.

Second, once the functions have been specified, the search for appropriate solutions can start. The final solution for the overall function is obviously not directly available and has to be created step by step, piece by piece. Hence, the role of the function structure is to guide the search for solutions. It enables problem decomposition and facilitates the recognition of parts for which the solutions are known or available. The level at which the decomposition has to take place depends on the level at which the search for solutions for each sub function seems most promising. When existing physical solutions can be assigned directly, the decomposition may end at a relatively high level. For totally new design, the decomposition has to be performed until levels of much lower complexity are reached.




▸ **Figure 2. 6** Function structure according to Pahl and Beitz

Third, once the functions are clearly specified, the search for solutions can be dealt with concurrently. A working principle has to be chosen for each function. A working principle expresses basic physical characteristics (geometry or material) to realize a physical effect that is needed to perform a given function. For instance, a working principle may be depicted as a rough sketch of a leverage that is based on the leverage law (physical effect) realizing the function 'amplify force'. Mapping each function in order to develop a working principle may be guided by a morphological scheme. For each function, collection of several alternative working principles is considered, from which the most appropriate one is selected.

Once a working principle has been chosen for each function, the challenge is to combine these working principles such that they together fulfill the overall function of the product. The combination of working principles is primarily based on the input-output relationship established clearly in the function structure. That is to say, each working principle has to realize its corresponding functional inputs and outputs. However, this is generally not sufficient. The compatibility of working principles is often strongly affected by physical and geometrical considerations. Alternative combinations of working principles have different effects on technical and economic criteria. Making a selection of physically feasible, and technically and economically favorable combinations is generally a hard task for designers. Taken together, the choice and combination of working principles results in a specification of an overall solution principle that is the starting point for the next stage.

*Embodiment design*
Designers will now proceed with the determination of the overall layout (construction structure) in line with technical and economical criteria. The physical parts are roughly arranged and the forms (shape and material) of the components determined. Designers have to check for function, strength, spatial compatibility, and financial viability. In addition, factors such as safety, ergonomics, production, assembly, operation, logistics, maintenance, recycling, and costs are considered. In dealing with all these aspects, designers will discover a great number of interrelationships, making iteration unavoidable. Pahl and Beitz aim to aid designers at this stage by providing a great number of checklists and guidelines. It is striking that they spend 74% of the pages in their book on describing embodiment design.

*Detail design*
During this phase, the arrangements, forms, dimensions, and surface properties are laid down in their final form. The materials are specified, production possibilities assessed, costs calculated, and all drawings and other documents are produced. Detail design has a major impact on production costs and quality, and therefore on market success.

The four stages outlined by Pahl and Beitz have now been described in sequential order. These stages may be regarded like the four runners in a relay race. Once the first stage is finished, the baton is passed to the next stage that in turn provides the trigger for the subsequent stage, and so on. It should be noted, however, that the metaphor of a relay race is only valid to a limited extent. In fact, Pahl and Beitz stress the iterative nature of problem solving.

In the first place, each stage includes steps where alternative actions are generated, tested, and selected. These cycles are each described within the specific context of the stage in which they should occur.

In the second place, cycles between the stages are also part of the potential of the problem-solving path such as is shown by the dashed lines in the illustration of the stages. Pahl and Beitz recognize and accept that these cycles may occur, but do not explore the cycles between the stages any further.

In the next section, the principles of axiomatic design will be described. Subsequently, VDI design and the axiomatic approach will be discussed and compared.

### *2.2.3 Axiomatic design*
Suh 1990, Albano & Suh 1992, and Albano et al. 1993 provide a general framework for structuring a design process called axiomatic design (AD). The key aspect is that designers are able to understand what the objectives of a product are, and the means by which these objectives are achieved. This framework is based on a fundamental set of design principles that (according to Suh) determine good practice. The axiomatic design group at MIT strongly advocates these principles and has devised many applications. Outside this group, axiomatic design is well known, but applied far less frequently. The abstract theoretical concepts require a lot of training and practice, and the strict rules probably constitute a stumbling block for many researchers. Nevertheless, many engineers refer to the basic principles and make use of various other aspects. The key elements are:

- *Domains*: The design process is modeled as the processing of information between the functional and the physical domain.
- *Hierarchy*: The design process progresses from a system level to more detailed levels. Decisions about the artifact are structured within a hierarchy in both domains.
- *Zigzagging*: The decomposition of the problem follows a top-down approach between the hierarchies of the two domains.
- *Design axioms*: These provide a basis for evaluating the design structure in order to realize good design quality and a smooth working sequence.

In general, axiomatic design develops products by continuously describing the functions and solutions within a set of constraints (Tate 1999). When the design process is initiated, the functions and solutions are described in a very abstract and simple manner. As the process evolves the product is illustrated in increasing levels of detail and by the time the product is finished every physical detail is known exactly.

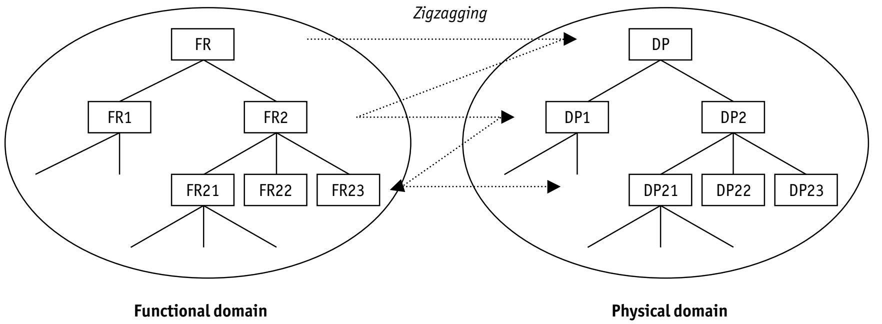
The unique feature of axiomatic design is that it provides a framework that shows in detail the interplay between the functions and solutions at every moment of the design process. More precisely, axiomatic design works with a system of functional requirements, physical design parameters and constraints. These will be defined below.

Functional requirements (FRs) are described as elements in the functional domain and represent design goals. Such requirements state what one wants to achieve and should be stated as a verb-noun pair without noting a particular solution. FRs can be decomposed at several hierarchical levels (see Figure 2.6), but at each specific level FRs are by definition independent of each other.

Physical design parameters (DPs) are elements of the physical domain and have the purpose of satisfying a particular FR. DPs describe how the FRs are achieved. That is to say, for each FR a DP needs to be selected in order to achieve the FR. The DPs can also be taken down to lower levels of abstraction. It is striking that the exact meaning of a physical DP depends on the hierarchical level in which it is considered. At the lowest hierarchical level, DPs become physical parts or very precise specifications of geometry, material, or tolerances (Hintersteiner & Friedman 1999). At higher levels, though, DPs are not necessarily physical and may represent general solution principles or concepts. The term 'physical' before the DP does not therefore imply that every DP is a piece or subassembly of a physical product (Albano & Suh 1992). A hierarchy within the physical domain is not by definition a hierarchy of a physical product describing the whole product, its building blocks, its subassemblies, its parts, etc. The term 'solution domain' would probably have been a much clearer indication that only the lowest level of the DPs refers to physical things that can be located somewhere on a physical product.

The constraints are specifications that the design solution must possess in order to be acceptable in the eyes of the customer and the designing or producing firm. In general, constraints depend on many decisions. Constraints impact on (multiple) FRs and limit the range of feasible DPs (Hintersteiner 1999). Recent work on AD (Tate 1999) has identified many different types of constraints. For the purpose of this thesis, however, the so-called global constraints that affect many design parameters and cannot be allocated to a particular function or solution (Albano et al. 1993) are the main ones under consideration. Examples of global constraints include restrictions in weight, size, or costs. The weight of the overall product for instance, is clearly a result of all components together and cannot be allocated to only one feature.

Having described the above constructs, the prescribed decomposition within axiomatic design will now be examined. Figure 2.7 shows that if FRs and DPs are decomposed, the graphic effect is a zigzag pattern. At the highest level of abstraction there is just one FR that needs to be satisfied by just one DP. The selection of an appropriate DP has the characteristics of a generate-test cycle (Albano & Suh 1992, Tate & Nordlund 1995, Tate 1999). Multiple DPs are generated and tested until a satisfactory one has been selected. Only if the FR has a corresponding DP can it can be decomposed into a set of sub-FRs. At this hierarchical level a sub-DP needs to be selected for each FR in a similar fashion to that just described. Again, once all sub-FRs are linked with a sub-DP, each sub-FR can be decomposed and the whole procedure repeats itself until the lowest level is reached (see e.g. Tate 1999).




▶ **Figure 2.7** Decomposition as zigzagging between the functional and physical domains

Axiomatic design pays considerable attention to the interplay between the FRs and the DPs. The so-called Axiomatic design matrix (A) indicates how the DPs together address the FRs at each level of the hierarchy. This is shown by the following equation: {FR}=[A]{DP} and illustrated by the three diagrams below. The structure of the axiomatic design matrix determines the sequence in which the product is designed. The matrix distinguishes between three different forms. It can be uncoupled (Figure 2.8), de-coupled (Figure 2.9) or coupled (Figure 2.10).

$$\begin{Bmatrix} FR21 \\ FR22 \\ FR23 \end{Bmatrix} = \begin{pmatrix} X & 0 & 0 \\ 0 & X & 0 \\ 0 & 0 & X \end{pmatrix} \begin{Bmatrix} DP21 \\ DP22 \\ DP23 \end{Bmatrix}$$

▶ **Figure 2.8** An uncoupled axiomatic design matrix

A design is uncoupled when only the diagonal elements (i=j) of the matrix have an 'X' and all others (I ‹›J) have an '0'. In that case each DP only impacts on its own FR and the DPs can be altered completely concurrently until each one is able to establish its FR. It should be noted that within the matrix all diagonal elements have by definition an 'X', since for each FR a corresponding DP has been chosen.

In practice, however, it is very likely that there are DPs that affect more than one FR. As a result the axiomatic design matrix also has non-diagonal elements filled with 'X's. A design is de-coupled when the matrix can be written in a lower triangular form, as can be seen in Figure 2.9. This implies that there is at least one DP that impacts on multiple DPs, and fully independent adjustment of DPs is not a realistic option anymore. Instead, the DPs need to be altered in a sequential fashion in order to achieve the FRs exactly. For the example shown in Figure 2.9, this means that first FR21 needs to be satisfied by DP21. Next, FR22 will be achieved by DP22 in collaboration with the already specified DP21. For FR23 it is the same story, but DP23 has to complete the effects of DP21 and DP22. Note that any other working sequence will result in unnecessary iterations.

$$\begin{Bmatrix} FR21 \\ FR22 \\ FR23 \end{Bmatrix} = \begin{pmatrix} X & 0 & 0 \\ X & X & 0 \\ X & X & X \end{pmatrix} \begin{Bmatrix} DP21 \\ DP22 \\ DP23 \end{Bmatrix}$$

▸ **Figure 2.9** A de-coupled axiomatic design matrix

Finally, a design is coupled when there are also relationships in the upper diagonal part of the matrix. In the example given in Figure 2.10, it can be observed that both FR21 and FR22 are uniquely fulfilled by DP21 and DP22. When the process starts by finding a solution for FR21, this solution needs to be altered in the case of FR2. When the designers start defining the right setting for DP21 and DP22 in order to achieve FR22, this in turn affects the performance of FR21. Quite clearly the resulting design process is highly iterative, and there is generally a long way to go before a setting for the DPs is found that is satisfactory for both F21 and F22. In Suh's opinion, a coupled design is an example of 'bad design' and should be avoided.

$$\begin{Bmatrix} FR21 \\ FR22 \\ FR23 \end{Bmatrix} = \begin{pmatrix} X & X & 0 \\ X & X & 0 \\ X & X & X \end{pmatrix} \begin{Bmatrix} DP21 \\ DP22 \\ DP23 \end{Bmatrix}$$

▸ **Figure 2.10** A coupled axiomatic design matrix

To sum up, the basics of axiomatic design have now been outlined. These will be helpful for the definition of product architecture but can also be used to analyze the product architecture in the later chapters of this thesis.

In the next section, the definitions of Pahl and Beitz, and axiomatic design will be discussed and compared in order to create a sound basis for defining and interpreting product architecture.

### 2.2.4 Discussion

This section compares the two engineering approaches described above. This is done for two main reasons.

The first reason is general scientific interest. It is very tempting to describe and compare two well-known methodologies that on the one hand have a similar goal of guiding the design process, but on the other hardly have any other aspects in common. Both have interesting and useful features and perhaps a thorough discussion of both paradigms will have future relevance. It is to be hoped that a detailed and precise discussion of their underlying rationale will prevent an indiscriminate combination of principles without any understanding of the exact differences between the two concepts.

The second reason refers much more directly to the central role of product architecture in this thesis. The definition of product architecture will be presented in the next section and will be based on a definition of functions and physical elements. In order to understand these definitions unambiguously and make further interpretation possible, the precise background(s) of these constructs needs to be described. The definitions of the two methods will thus be considered synonymously, and comments about their differences made. After the main constructs of product architecture have been presented in the next section, the discussion here will be revived and a choice made about the definitions that will be used during the remainder of this research. Having good definitions and, moreover, knowing what prescriptive approach can be referred to in order to analyze the product architecture, will provide a sound basis for the research.

At a general level, VDI provides an illustrative overview of all steps that need to be taken during a design process. These steps are easy to understand and suggest effective ways of working towards a final solution. However, VDI does not really focus on how decisions need to be made and what the potential consequences of particular decisions are. That is to say, while the whole approach is largely based upon the principles of decomposition, it does not indicate what 'good' or 'bad' decompositions are in a specific situation.

On the other hand, AD provides a clear focus on the underlying decomposition of a product and continuously indicates the effects of a particular design decision on the overall structure of the design process. However, the axiomatic design approach is very conceptual and surrounded by strict rules. A well-trained eye is needed to fully envisage a product within the framework of multiple levels of abstraction.

Compared with AD, VDI does not explicitly consider the functional and physical domains or the mapping characteristics they have in common. At first glance, though, the function structure is similar to the functional domain, and the concepts of working principle and working structure come close to the physical domain (Tate 1999). Furthermore, both approaches describe the role of generate-test cycles in finding a solution for a specific function. The following differences will be discussed:

- The definition of a function.
- Types of functions.
- Decomposition of a function.
- The combination of physical solutions.

*The definition of functions*

VDI and AD have in common that a function has to be specified as a verb-noun pair, preferably in a solution-neutral manner. However, AD has a much broader definition of functions than VDI.

AD defines functions as a design goal, or what needs to be achieved. VDI does not contradict this but has a more specific (narrow) definition. VDI defines functions as the transformation of material, energy or information. This implies that important design goals such as aesthetics cannot be captured within the function structure of VDI since it is barely possible to describe them in input-output language. As a result, important design goals are not captured within a VDI function structure at an early stage of design, but come in much later in the design process.

The point is that while every VDI function can be considered a goal (AD), not every goal can be described as a transformation of energy, material, or information.

This research suggests that it would be wise for VDI to take important design goals into account as early as possible in the design process and not to wait, because these cannot be illustrated as a transformation of material, energy, or information.

*Different types of functions*

VDI draws a distinction between main functions and auxiliary functions whereas AD considers all functions as essentially the same. The VDI distinction is based on the question of whether a function makes a direct or indirect contribution to the overall function. However, why Pahl and Beitz label one function differently than another is a question that perhaps needs to be asked. The answer is probably not that one is more important than the other since both types are eventually needed to fulfill the overall function. To return to the previous example, it is quite clear that if the PC cooling system doesn't function, the computer will soon stop working. Hence, both the auxiliary and the 'main' functions are essential. The only reason that can be suggested is that the difference is based on a preconceived sequence of working, as the following would suggest:

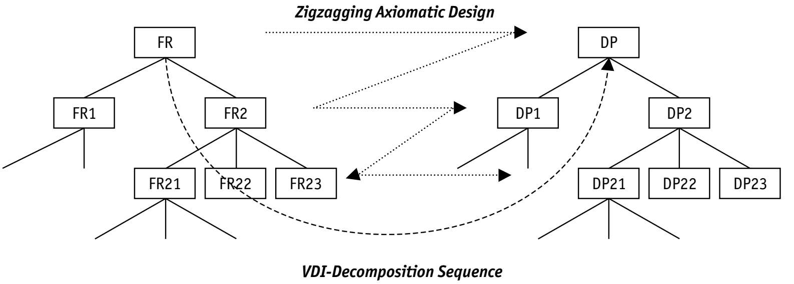
> *Auxiliary functions have a supportive or complementary character and are determined by the nature of the solution. (p.32) It is useful to start with determining the main flows in a technical system. The auxiliary flows should be considered later. (p.156) The auxiliary flows help in the further elaboration of the design in coping with faults, and in dealing with problems... (p.156) The complete function structure, comprising all flows, can be obtained by iteration, ...first...the main flow...completing that by the auxiliary flow...and then establishing the overall structure. (p.156)*

Alternatively, within AD the sequence of working is based on the structure of the AD matrix, or based upon the top-down principles of the problem-solving hierarchy. Put differently, the design process generally starts with a parent FR and once the solution is found it proceeds with child FRs (Hintersteiner 1999). Within the same hierarchical level, the AD matrix determines the (optimal) sequence of solving the FRs.

Taking this into account, it is at the very least remarkable that auxiliary and main functions are considered with same degree of detail in view of the fact that the cited passages state that the auxiliary function is fully specified by the solution of the main function. This apparently suggests that auxiliary functions may be considered on a lower hierarchical level than the main function (according to zigzagging. This will be discussed below). There would thus appear to be no real reason for Pahl and Beitz to have identified two types of function (see also Tate 1999).

*Decomposition*

AD and VDI differ theoretically in how they perform a decomposition. AD prefers the zigzagging approach and VDI first makes a fully functional decomposition that one assumes is solution-neutral. Only when all sub-functions are completely specified does VDI initiate the search for sub-solutions. Subsequently these sub-solutions are abstracted to the whole again. Figure 2.11 outlines how the various decomposition sequences operate. For simplicity, only FRs and DPs are plotted.




▸ **Figure 2.11** The decomposition strategies of AD and VDI compared

This shows that Pahl and Beitz decompose in an U-shape, first a full decomposition of the overall function and then a bottom-up approach to reach the final solution. On the other hand, AD communicates between the functional and physical domains at each level of the hierarchy.

Having made this difference clear, it should be noted that Pahl and Beitz deviate considerably from their prescribed decomposing sequence in their examples. Among other things, they make the following remark:

> *It should be remembered that function structures are seldom completely free of physical or formal pre-assumptions. Hence, it is perfectly legitimate to conceive a preliminary solution and then abstract this by developing and completing the function structure by a process of iteration.' (p. 160)*

In conclusion, it is reasonable to assume that a function cannot be decomposed without considering its solution (on that hierarchical level). This is a very important concept and one that will be used again in Chapters 4 and 5.

*Combining physical solutions*

VDI and AD both aim to search for a solution for each function (at a particular level). However, the way that all of the solutions together fulfill the overall function is modeled differently.

The advantage of VDI is that its functional scheme provides a clear overview of the inputs and outputs that each working principle has to realize in order to obtain correct functioning of the whole., The DPs within AD also obviously have to realize the correct specifications of the FRs, but how all these flows are connected is not clearly visible.

The drawback of VDI is that it does not model interdependencies between working principles due to reasons other than functional inputs and outputs. A working principle that affects more than one function at a time, for instance, cannot be captured within the VDI function structure. Pahl and Beitz argue that technical and economic reasons affect the arrangement of working principles, but do not formalize these nor give any indication of how to deal with these interactions.

The advantage of AD is that it formally models how a set of FRs relate to a set of DPs and what the implications are. This is clearly visible when a DP affects multiple FRs. There is an ongoing focus on 'good' or 'bad' decompositions as early as possible in the design stage.
The following can thus be concluded:

- AD defines functions as design goals and is more general than a transformation of energy, material, or information (the VDI function). However, the function concept of VDI may be considered a sub-set of the AD definition of functions.
- Pahl and Beitz distinguish between main and auxiliary functions but no convincing reason for the necessity of this division can be detected.
- A close look at both approaches indicates convincingly that a function cannot be decomposed without addressing its solution.
- The VDI functional scheme provides a clear overview of how the inputs and outputs interact, and AD shows how the mapping between functions and solutions is established. Both elements are important.

The next section explores the definition of product architecture. This will be followed by a return to the above discussion and a set of definitions that will be used during the remaining research will be devised.

## 2.3 Product architecture

Having developed a working idea of the main decisions that take place during a design process and having discussed the basic technical constructs (functions and solutions), the time has come to define product architecture. In broad terms, product architecture can be defined as the way that distinct product building blocks interact in order to obtain correct functioning of the whole. Section 2.4 will illustrate that product architecture has an enormous impact on the manufacturing firm, and is deeply embedded in a company's way of working. The present section will initially address the definition of product architecture and its characteristics in considerable technical detail based on the work of Ulrich (1995).

First, the definition of product architecture will be defined and the three technical decisions that determine product architecture explored. Ulrich's views on defining functions and solutions will be described, and the discussion will conclude with a clear set of definitions that will be used during the remainder of the research.

Second, product modularity, an essential characteristic of product architecture, will be examined.

Third, the means available to represent a particular product architecture in a fashion such that it can be analyzed will be discussed with reference to the work of Pimmler and Eppinger (Pimmler & Eppinger 1994) who model the interactions between building blocks.

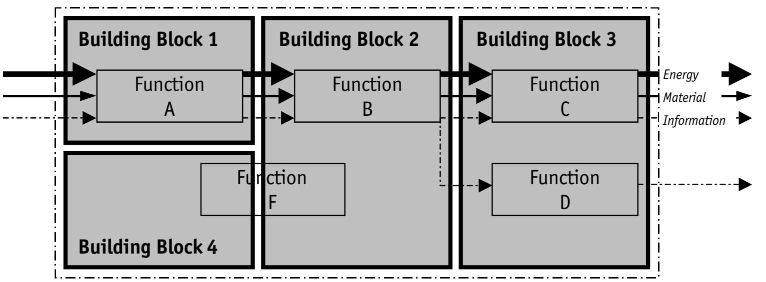
Even though this section is relatively small and rather technical, it is essential for the study as a whole. Not only the technical decisions relating to architecture but also the modeling of interactions will form the basis for a linking of product architecture to organization in Chapters 4 and 5.

### *2.3.1 Definition of product architecture and modularity*

In the foregoing sections it was observed that there is a myriad of research methods and terminology within engineering design literature. Definitions concerning product architecture form no exception to this (Pahl & Beitz 1996, Erixon 1998, Stake 1999, Blackenfelt 2000). However, instead of analyzing the entire set of alternative definitions of product architecture, the basic assumptions and definitions of Ulrich (1995) will be largely adopted. He managed to combine elements from several schools and proposed a technical definition of product architecture that is widely accepted in the literature.

In order to define product architecture, Ulrich basically considered the functions and the physical building blocks of a product. He proposed three technical decisions that together form the architecture of a product. These are:

- The arrangement of functions.
- The mapping from functions to physical building blocks.
- The specification of the interfaces among interacting physical building blocks.




▶ **Figure 2.12** A conceptual illustration of product architecture

### *The definitions of functions and building blocks*

Before these decisions can be further described, the constructs of functions and physical building blocks according to Ulrich must first be defined. They can then be integrated into the previous discussion.

Ulrich states that a function is what a product does as opposed to its physical characteristics. Functions can be arranged by linking their inputs and outputs (of energy, material, and information). In addition, Ulrich recognizes that not all functions can be included within an input-output language. Ulrich and Eppinger 2000 recently stated the following:

> *Also note that in some applications the material energy, and signal flows are difficult to identify. In these cases, a simple list of sub-functions of the product without connections between them is sufficient.*

Ulrich defines physical building blocks as physically distinct chunks of a product. The building blocks of a computer, for instance, are the monitor, keyboard, hard-disk, etc. A product consists of building blocks that are in turn each made up of physical elements or components. Physical elements include the keys, the board, the switches, the LEDs, etc. These elements include the characteristics that are required to fulfill the functions.

People often talk about modules instead of building blocks. However, that is generally avoided within the literature since this has an implicit association with modular and not every building block is modular.

Based on the previous discussion in section 2.2.4 the constructs that will be used during the remainder of this research will be outlined below.

- *A function* is a *design goal*, describing what needs to be achieved (in line with AD). If it is possible, functions are conceptualized as transformations of energy, material, or information (in line with VDI). No distinction is made between main and auxiliary functions.
- *Building blocks* are purely physical and are not the same as the physical domain within AD, or working principles. However, it can be said that building blocks implement working principles or locate design parameters. Since at the lowest level of the physical domain DPs become physical things, it can be stated that each building block locates a whole collection of DPs (at the lowest level of abstraction) that ultimately fulfill the overall function of the building block.

Together these definitions allow an interpretation of product architecture based on the detailed considerations of the previous chapters. All the definitions are in line with axiomatic design, and the clarity of Pahl and Beitz has been added where possible. Ulrich's three architectural decisions will now be described.

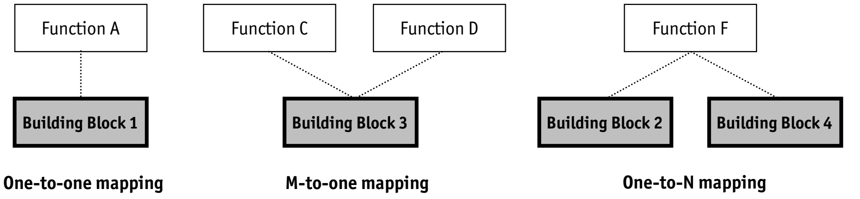
*The arrangement of functions*

Figure 2.12 shows an arrangement of a product's functions. It shows that functions A, B, C, and D are connected by flows of energy, material, or information. Function F is a design goal that cannot be captured by input-output language.

*Mapping from functions to building blocks*

The bold boxes in the same diagram indicate the physical building blocks of a product. These blocks each implement the functions. Block 1 implements function A, block 3 achieves functions C, and D, and function F is established by blocks 4 and 2 together. An important characteristic of architecture is thus its mapping, by means of which building blocks are connected to a function or functions.

Possible mappings between functions and building blocks are one-to-one, many-to-one, and one-to-many such as is depicted in Figure 2.13 (where M=N=2). A combination of one-to-many, and many-tone results in a many-to-many mapping (M-to-N). Note that for clarity the functions and the building blocks refer to the same level of abstraction.




▸ **Figure 2.13** Three types of mappings between functions and building blocks

*Physical interfaces and coupling*

Physical interfaces between interacting building blocks enable the physical realization of exchanges of energy, material, and information, and /or their geometric connection. A keyboard and a personal computer, for instance, are connected by a cable that realizes the exchange of information and energy. Two blocks can have a physical interface even when they do not exchange any flow. A plastic bottle and its cap have an interface but no exchange of material, information, or energy (except if one decides to model all forces between the two blocks, which is not common practice).

Physical interfaces cause a certain coupling between the blocks. The geometry of the cap, for instance, cannot be endlessly adjusted without changing the geometry of the bottle. Ulrich presents the following definition:

> *An interface is coupled if a useful change to one building block affects a change to the other block in order for the overall system to work correctly.*

Note that this definition may refer to anything. However, a closer reading of the work of Ulrich suggests that the coupling refers to 'physical' aspects. Based on the examples in the paper, three reasons for coupling can be identified:

- A geometric connection that hampers the geometric freedom of both blocks.
- Side effects such as vibration, heat or magnetism of one block that effect the other block.
- Limited space that limits the freedom of size of both blocks.

Within product architecture the amount of coupling is an important feature. Interfaces may be coupled or 'de-coupled'. Coupling is obviously a relative property. It is only possible to say that one interface is more coupled with respect to a particular change than another interface. For example, a mouse that is connected by a cable to a personal computer is more coupled then a cordless mouse since the latter is much less limited with respect to the position between the mouse and the computer.

Furthermore, it is worth emphasizing that coupling depends on what is considered a useful change. If it were feasible to control a computer with a cordless mouse at a distance of 100 kilometers then the cordless mouse would have a coupled interface. As things stand at the moment, however, this is not relevant.

To sum up, coupling refers to anything that constrains the design of another building block, though it is advisable to only consider relevant constraints.

*Modular versus integral*

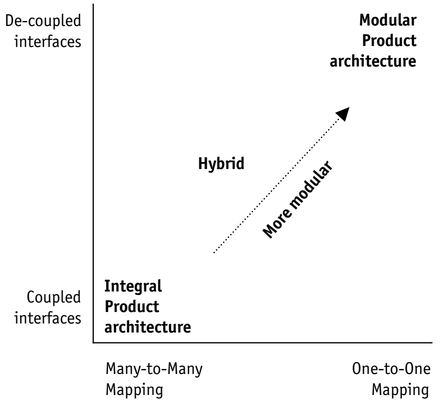
Perhaps the most important property of product architecture is the dependence between its building blocks. To reduce things to the simplest terms, there are two types of architecture: modular and integral.

When building blocks are completely independent of each other the product is 'modular'. On the other hand, a strong dependence between the blocks corresponds to an 'integral' architecture. In the most radical situation, a completely integral architecture will in fact include a product that cannot be decomposed into building blocks. All its pieces are equally strongly dependent. An ultimately modular product would actually consist of a collection of fully separate blocks that form no whole. In practice, however, there are hardly any such extreme cases and modularity is a relative property. For instance, a standard personal computer is more modular than a hand-held computer, and an SLR camera is more modular than an instant camera (for examples, see Ulrich & Eppinger 2000).

Ulrich defines modularity very precisely based on the previous described features: mapping and interface coupling.

- A modular product has a one-to-one mapping between functions and building blocks, and its interfaces are de-coupled.
- An integral product has complex (N-to-M) mappings between the functions and the building blocks, and its interfaces are coupled.

In general, products are neither entirely modular nor integral and lie somewhere between the two extremes. These are called hybrid architectures, and they have few non one-to-one mappings and some coupled interfaces. As a rule, the more the mapping is one-to-one and the more the interfaces are de-coupled, the more modular a particular product architecture is. This is illustrated in Figure 2.14.




▶ **Figure 2.14** The range between modular and integral architectures

### *2.3.2 Representing architecture*

Having addressed the basic architectural decisions, the question of how a particular architecture of a product can be represented in sufficient levels of detail will now be considered. An answer is required since this research aims to analyze a design process based on an understanding of the underlying product architecture. This section will therefore explore whether the available literature provides a means for clearly representing a particular product architecture. In particular, the work of Pimmler and Eppinger (1994) who proposed a taxonomy of interactions that is commonly used to model the architecture of a product will be explored. This approach will first be described and then discussed.

#### *Architecture as interactions between building blocks*

The starting point is the assumption that the above definitions of product architecture and modularity are very similar to Simon's hierarchical concept as described in section 2.1. Modular products can be compared with the concept of nearly decomposable systems where the interactions within the blocks are much stronger than the interactions between the blocks. The more integral an architecture, the stronger the interactions between the blocks become.

Based on this, Pimmler and Eppinger proposed a method for constructing a product architecture by illustrating the interactions within a product. They documented all physical elements (N.B. these are decomposed building blocks) of a product and identified all their interactions. They subsequently modeled all elements and interactions in a matrix – the elements were plotted identically in the rows and columns of the matrix and the interactions between them in the elements of the matrix. The search for relatively autonomous groups of physical elements then began. Their aim was to identify clusters of physical elements where the interactions within the clusters were much stronger than the interaction between the clusters. Once satisfactory clusters of physical elements had been found, each cluster formed one of the building blocks of the product. The matrix thus shows the building blocks of a product and their interactions.

Especially interesting is the way Pimmler and Eppinger defined interactions between building blocks. They recognized that building blocks may have different types of interactions since there are many different technical reasons behind a particular product architecture (as Ulrich demonstrates). Accordingly, they proposed a taxonomy of four types of interactions.
The following four types were identified:

- *Energy*: An energy-type interaction identifies the need for adjacency or orientation between two building blocks (or physical elements).
- *Material*: A material-type interaction identifies the need for materials exchange between two building blocks (or physical elements).
- *Information*: An information-type interaction identifies the need for information or signal exchange between two building blocks (or physical elements).
- *Spatial*: A spatial-type interaction identifies the need for adjacency or orientation between two building blocks (or elements).

Each interaction type can be rated as required, desired, indifferent, undesired, or detrimental for functioning. It should be noted that the first three types refer to *physical* exchange of energy, material, or information, respectively. For instance, the driving unit of an electric shaver requires energy (electricity) from the power supply to perform its function of providing rotation. To physically realize this, an electric wire must link both blocks. An example of a detrimental exchange of energy is a driving unit that hampers the functioning of the power supply because it vibrates too much. Other reasons include too much heat or radiation.

The spatial interaction type concludes the taxonomy with an indication of whether blocks should be located in proximity to each other, as far from each other as possible, or somewhere in between.

The work of Pimmler and Eppinger can be viewed in two ways. First as a promising way of constructing new architectures, and second as a clear way of visualizing a particular (existing) architecture. We are mainly interested in the second aspect, which will be discussed below.

*Discussion*

In order to come up with a clear way of representing product architecture, the matrix representation and the taxonomy of interactions will be briefly discussed.

– *Matrix representation*

The matrix representation offers a clear way of modeling interactions between building blocks. Every interaction between each possible pair of building blocks can be captured and, moreover, each possible type of interaction can be modeled. The traditional way of illustrating architecture (see Figure 2.12) is much more limited. In such a scheme of blocks and functions it is very difficult to show the interactions between all possible pairs of blocks, and not all interactions (i.e. the interfaces) are visualized. The modeling of interactions and building blocks thus provides the perfect basis for depicting and analyzing a product architecture.

– *The taxonomy of interactions*

The proposed taxonomy of interactions has often been applied in recent research and serves as a basis for architectural studies (Terwiesch & Loch 1999, Stake 1999, Blackenfelt 2000, Sosa et al. 2000). In order to obtain an initial idea of whether it is suitable for this research, the taxonomy will be compared with alternatives in the literature. The taxonomy will then be compared with the work done by Ulrich.

Table 2.1 illustrates several alternative taxonomies that have appeared in the literature. It is not the goal of this research to elaborate these in great detail, but it should be observed that the taxonomies have much in common. They all draw a basic distinction between exchanges between the three flows (energy, material, or information) and geometrical aspects, including undesired or unintended effects or side effects. The aspects addressed within the taxonomy of Pimmler and Eppinger are thus almost identical to the issues that have been distinguished by other researchers in the field. However, compared to the others, it seems that Pimmler and Eppinger pay slightly less attention to the geometrical aspects of their interaction types.

▸ **Table 2.1** Different types of interaction (based on Stake 1999)

| *Pimmler and Eppinger 1994* | *Ulrich and Eppinger 2000* | *Erixon 1998* | *Hubka and Eder 1988 (Sanchez 1999b)* |
|---|---|---|---|
| Physical Exchange:<br>Material<br>Energy<br>Information | Fundamental:<br>Material<br>Energy<br>Information | Energy | Power,<br>Signals |
| Spatial (=Proximity) | Incidental:<br>Geometric arrangement+ | Geometry | Spatial<br>(location + volume)<br>Attachment |
| Detrimental:<br>(side effects) | Side effects | | Environmental<br>(Side effects) |

In the next section, the taxonomy of interactions will be discussed further and the taxonomy compared with the definition of product architecture.

### *2.3.3 Comparison*

A comparison of the previous two sections (2.3.1 and 2.3.2) leads to a remarkable and important observation. The decisions that according to Ulrich determine a product architecture, and the two important features, mapping and interface coupling, are in fact difficult to capture within Pimmler and Eppinger's four types of interaction. The most compelling question is how can taxonomy include a function that is mapped across two building blocks? A short glance at the taxonomy will probably be sufficient to show that this is not possible.

This concept is an important one in this thesis and will be further elaborated in Chapters 4, and 5. There, the benefits of being able to link the architectural three decisions with interaction types will become apparent, and a new taxonomy of interactions will be developed. This topic will not be pursued in the present section since the organizational principles also need to be taken into account when discussing the taxonomy.

At this point it will suffice to state that despite the whole engineering world being convinced of the validity of Ulrich's architectural decisions, these are not clearly present within taxonomies that have been proposed to represent product architecture.

To sum up, the following points can be made:

- A matrix representation that illustrates building blocks and their interactions seems a very useful way of clearly representing an architecture.
- Since product architecture consists of several technical decisions there will be various types of interaction between building blocks.
- The available taxonomies in the literature are quite similar and cover a broad range of technical aspects, but have a remarkably poor correspondence with the architectural decisions that determine a particular product architecture.

## 2.4 The implications of architecture

The previous sections have gradually led to a definition of product architecture. Now is the time to finally demonstrate why product architecture is such a crucial issue. A brief summary of available studies that have extensively reported on the high impact of architecture on manufacturing firms will now be given. Furthermore, the reasons why some products are modular and others integral will be listed.

Product architecture has been a frequently studied topic in relation to many business topics. Many have produced very complete overviews of all implications and considerations. Ulrich 1995, Ulrich and Eppinger 1995, and Ulrich and Eppinger 2000, for instance, suggest that product architecture affects how variety is established within production, how change can be realized across subsequent generations of products, how building blocks can be standardized, the overall performance of products, and the management of product development. On the other hand, Erixon 1998, Stake 1999, and Blackenfelt 2001 argue that architecture impacts on product development, product variants, quality (testing), purchasing and after sales.
It is recognized by all authors that the design of modular products has a number of clear benefits.

- Modular products allow the production of a great variety of end products from a limited number of building blocks (Ulrich 1995, Erens 1996, Martin and Ishii 2000).
- Modular products allow for a platform strategy permitting a great number of new variants to be developed based on a stable architecture (and few standard building blocks) (Wheelwright & Clark 1992, Meyer & Utterback 1993, Sanderson & Uzumeri 1995, Meyer et al. 1997, Robertson & Ulrich 1999).
- Modular products facilitate changes to products once introduced (Baldwin and Clark 1997, Sanchez 1999c).
- Modularity simplifies parallel testing and maintenance (Ulrich 1995, Ishii 1997, Erixon 1998).
- Modularity allows for parallel development of design teams (Sosa et al. 2000, Baldwin & Clark 1997, Ulrich & Eppinger 2000, Blackenfelt 2001).
- Modularity allows for outsourcing of building blocks (Novak & Eppinger 1998).
- With today's pressure and increasing complexity there is clearly a trend in favor of more modular products (Baldwin & Clark 1997).

On the other hand, modularity also has a number of fundamental limitations or drawbacks:

- Too much modularity can make products look too much alike (Cutherell 2000).
- Modularity increases the risk of competitors copying the design (Cutherell 2000).
- Modularity is generally at the expense of unit cost and increases the volume (size) or weight of the product (Ulrich 1995, Whitney 1996).
- Modularity may be limited by the technology available (Whitney 1996).
- Designing modular products may be very difficult, be initially time-consuming (Ulrich, Sartorius, Pearson, Jakiela 1993), and depend on the capabilities of available designers within the company (Meyer & Utterback 1993).

Studies done by Henderson and Clark (Henderson & Clark 1990) (which will be returned to later in this thesis) prove that an established architecture is strongly embedded within the organization and the company's way of working. In addition, Sanchez (1999a) argues that the entire 'knowledge' of a firm is strongly shaped by the architecture of a product.

Based on this brief summary of factors that have to be addressed within the context of product architecture, it is reasonable to state that architecture strongly affects and is affected by a firm's strategic considerations, and furthermore heavily influences how the company actually works.

Later in this research this knowledge will prove invaluable in understanding the causes of particular architectural decisions, but also in understanding that changing an architecture is not just a matter of techniques, but also affects the firm's entire policy.

## 2.5 Summary

This chapter's objective was to define the concept of product architecture and its underlying decisions, and to explore how a particular product architecture can be represented. Product architecture was defined according to the work done by Ulrich. Product architecture is:

- The arrangement of functions.
- The mapping from functions to physical building blocks.
- The specification of the interfaces among interacting physical building blocks.

The most important characteristic of product architecture is its amount of modularity:

- A modular product has one-to-one mapping between functions and building blocks, and its interfaces are de-coupled.
- An integral product has complex (N-to-M) mappings between the functions and the building blocks, and its interfaces are coupled.

In general, products are neither entirely modular nor entirely integral but rather somewhere in between the two extremes. They are called hybrid architectures. As a rule, the more the mapping is one-to-one and the more the interfaces are de-coupled, the more modular a particular product architecture is.

Furthermore, the importance of definitions of functions and building blocks in order to obtain a correct understanding and interpretation of the definition of architecture was stressed. The design methodology of Pahl and Beitz (1996) and the Axiomatic Design approach of Suh (1990) were discussed and their definitions compared. The following set of definitions was produced and will be used during the remainder of this research.

- A function is a design goal, describing what needs to be achieved (in line with AD). Where possible, functions should be conceptualized as transformations of energy, material, or information (in line with VDI). The VDI function concept is in fact a sub-set of the AD definition of functions.
- Building blocks physically fulfill functions. They locate a collection of detailed physical design parameters that specify the physical properties required to establish the functions.
- In order to find a solution for a function, generate-test cycles are required.
- A function cannot be decomposed without addressing its physical solution (at the same level of abstraction).
- Constraints are specifications that limit the range of possible solutions. Global constraints depend on many decisions and cannot be allocated to specific building blocks or functions.

These definitions and the engineering design models described in this chapter will provide the basis for the interpretation of product architecture in an organizational context.

It was then argued that a particular architecture can be clearly presented as a set of interacting building blocks depicted in a matrix. Various technical reasons why building blocks may interact are captured within a taxonomy of interactions that distinguishes several types of interactions. The best known is the taxonomy by Pimmler and Eppinger who propose 4 types of interactions. It was, however, concluded that this taxonomy has a poor correspondence with the original definition of product architecture by Ulrich. This aspect will be revisited in Chapters 4 and 5.

Finally, the underlying contingencies of product architecture were described. This knowledge is valuable for understanding the causes of particular architectural decisions, but also for understanding that changing an architecture is not just a matter of techniques but also affects a firm's entire policy.

# 3 Organization of design processes: principles and applications

This chapter addresses the organization of design processes and is the second and final step needed to link product architecture with organization in Chapter 4.

It is generally recognized that product development is not usually a one-man job but rather an interplay between many people that each add a small piece to the overall 'puzzle' of creating a new product. It is therefore crucial for a successful design project that these people work together in an effective manner and collectively complete the job as a whole. Effective coordination between the designers is not something that comes out of the blue. It is obvious that if during a design project all designers have complete freedom to do what they want and in complete isolation from each other, the outcome will certainly not be a product with the proposed specifications, not to mention the quality, time and cost issues. Hence, what is needed in every project is some kind of structure that determines what task has to be done by each person, and that specifies how all these tasks need to be integrated. Within organizational science the question of what effective organizational structures are has attracted scholars for decades and resulted in a number of organizational principles.

In this chapter, two key issues that heavily impact on organizational practice and are the foundations of organizational theory will be examined. They are (1) the division of work into tasks, inevitably followed by (2) the coordination of those tasks to accomplish the overall operation. To obtain a general view of these features, there will be a brief examination of classic organizational theory in the first section. Some classic principles that are extremely valuable within the context of product development and this research in particular will be listed. The second section shifts attention towards the organization of design processes and focuses on so-called Design Structure Matrix (DSM) studies that analyze the existing organizational structures of design projects and suggest means to improve them. Later in Chapter 4 these models will provide important input bridging the gap between architecture and organization.

The emphasis of this chapter is on structures of organization. People work in organizations and there are an enormous number of human aspects that may affect collaboration between people. Despite this, behavioral perspectives will be considered only indirectly and play a secondary role here. The following two issues will be addressed:

- Classic organizational theory.
- Organization of design processes.

## 3.1 Classic organizational theory

This section will explore classic organization theory. After a general introduction, the theories of Galbraith (1973) and Thompson (1967) will be examined. Some alternative insights will then be dealt with, followed by a critical discussion of the work of Thompson. Finally, an overview of the classic theories will be presented, as well as this thesis's interpretation of them, which will inform the remainder of this research.

### *3.1.1 Introduction: tasks and coordination*

Organizational theory has for a long time been based on a paradigm that conceptualizes organizations as systems that process information within uncertain environments (Nonaka 1994). Organizations handle information in order to achieve a business goal that is assumed to be clearly defined and not to shift over time.

Many researchers have addressed the question of how the overall operation of a company can be divided into smaller information-processing tasks that can be allocated to human resources. An organization is assumed to be a collection of tasks that process and exchange information and together achieve the overall business goal or goals. Tasks may need information from other tasks and generate information that may be required for alternative tasks.

The main reason why an overall job is decomposed into smaller tasks is that this is an efficient way of dealing with the bounded rationality of the organizational members (Thompson 1967, Simon 1981). A single person simply does not have the skills and time to achieve the overall objectives of a company by himself. By defining smaller tasks people can focus their skills and talents on a specific piece of work. Broadly speaking, it is generally assumed that the more specialized a person is at a particular task, the better the task performance. Consequently, special jobs are isolated and assigned to individuals or specialized units, resulting in a kind of departmentalization of the company (Mesarovic et al. 1970, Mintzberg 1979).

This division of work inevitably makes coordination of all tasks of major importance. Fragmented pieces of work need to be coordinated in order to accomplish the overall operation. Put differently, coordination is required to compensate for the fact that tasks are performed as if they are fully independent (Mesarovic et al. 1970).

Organizations are challenged to create mechanisms that permit coordinated action across large numbers of interdependent people. Our understanding of effective coordination has its roots in ancient times, as the following biblical quotation indicates (Bratton 2001):

> *Moses' father-in-law advises Moses…you will not be able to perform it yourself alone…you should provide out of all people available men…and place such over them, to be rulers of thousands, rulers of hundreds, rulers of fifties, and rulers of ten…and every great matter they shall bring to you, but every small matter they shall judge*

From pyramid to computer, organizations have always aimed to find ways of handling the coordination problem, all of which manage to include the ingredients found in this quotation. In this section, some fundamental organizational theories (from the nineteen-sixties) will be explored further. The first thing to do, however, is to delineate the relevant area of organizational theory.

This research focuses on theories directed at organizational structures that facilitate coordination of large numbers of people. It has been pointed out that every organizational structure has to deal with specialization, but also has to manage the interdependencies between all of the tasks. Logically, therefore, there are two structural aspects that affect coordination – the number of tasks, and the interdependence between the tasks.

Both aspects are important variables in the identification of organizational structures that simplify coordination. However, as will be argued below, the variability of interdependencies is most obvious within the context of product development.

As coordination is a necessary consequence of specialization, the need for coordination can be reduced by lowering the level of specialization. Hence, the less divided the tasks, the less effort is needed to integrate the tasks into a whole. For many industries within production environments this has led to a trend of less specialization in low-to-medium skill tasks. Low-skill tasks have been automated and employees have moreover been trained to be able to perform broader (less fragmented) tasks. These new arrangements offer higher motivation and lower coordination effort, which generally compensates for the loss of specialization (Galbraith 1994).

For the product development of complex products, though, work still must be assigned to experts who have in-depth knowledge of specific areas. Innovative companies have high (and even increasing) levels of specialization, which makes a high focus on interdependencies indispensable. Lowering the number of tasks is simply not a feasible option in these cases.

The following sections shall therefore address the perspective of interdependencies between tasks. Two classic theories that propose several coordination mechanisms to handle interacting tasks will be described and moreover a policy of splitting up the work into semi-autonomous groups in order to facilitate coordination will be proposed. First, Galbraith's theory will be described in order to provide a general view of the information processing model and obtain some useful insights. Second, the work of Thompson, who was much more precise in the distinctions he made between the various types of interdependencies and corresponding coordination mechanisms, will be described.

As the next step, some alternative ways of structuring organizations will be discussed. Finally, there will be a critical review of the work of Thompson, and a summing up.

### *3.1.2 Structuring organizations according to Galbraith*

In his book "Organization Design", Galbraith (1977) provided an insightful view of the coordination problem, one that is largely in line with the advice to Moses. A brief overview of his assumptions will be given and then a report on his ideas.

Galbraith assumes that human resources have a limited capacity to process information. He also believes that organizations have to deal with uncertainty in the sense that they do not have enough information to perform tasks to a specific level of performance. The greater the uncertainty of the task, the greater the amount of information that must be processed by decision-makers during the execution of that task. Uncertainty causes unanticipated events. The more uncertainty, the greater the number of non-routine events that cannot be anticipated or planned for. Galbraith's central idea is that the greater the uncertainty, the more difficult it becomes to achieve a nicely coordinated role. The key to an effective organizational structure is being able to cope with exceptions.

Galbraith identifies three ways of achieving coordination – coordination by rules or programs, coordination by hierarchy, and coordination by targets or goals. The order that they are implemented corresponds to their appropriateness as a means of handling uncertainty.

To look first at rules, programs, and procedures, these can be used to specify how individuals within the company should behave or act. If everybody sticks to the rules and do what they are required to do in a specific situation, the company will function in a well-coordinated way. However, difficulties arise when a new situation appears that is not covered by the procedure.

Second, Galbraith suggests giving some workers a managerial role, comprising a number of tasks (and people). In unique situations, a person can call upon a person at a higher hierarchical level who has enough information to make the decision to tell his subordinates what to do. However, when too many unforeseen events arise, there is an inherent risk that such a managerial hierarchy will become too unwieldy.

Third, targeting or goal-setting is described. The essence of targeting is that an individual or unit is told to do the job no matter what it takes. In effect, individuals are given the power to deal with all events, unforeseen or otherwise, as long as they achieve their goal. The assumption is that when every person or unit realizes his or her goals, the overall goal of the company will be achieved and management will not be overloaded by details. Management is also there to deal with the situation should people fail to meet their goals and new courses of action need to be decided on. This is similar to the advice given to Moses in the quotation above.

However, the above-mentioned coordination mechanisms alone are not sufficient within uncertain situations since the risk of management overload is still present.

In addition to the coordination mechanisms, Galbraith also introduced four strategies. The first two in fact refer to reducing the need for coordination, and the second two yield mechanisms for increasing the ability to process information. Briefly, the strategies include:

- *Creation of Slack Resources*: The probability of someone failing to meet a target is reduced by making the goals 'easier' to achieve, for instance by extending delivery times, or adding more money to the budget.
- *Creation of self-contained Tasks*: When the work of several units is relatively independent of others, less coordination is required between the units. Should one fails to meet its targets, it only affects the others to a limited degree.
- *Investment in Vertical Integration Systems*: Condensing the flow of information by building specialized languages and computer systems can help analysis and decision making.
- *Creation of Lateral Relationships*: Moving the decision-making power down in the firm to where the information exists can reduce uncertainty at the decision level. Various strategies of increasing complexity can be employed to achieve this – direct contact between managers across groups, liaison personnel between groups, task forces, teams, etc.

The four strategies will reappear, both implicitly and otherwise, in this research. The main lesson to be learned from Galbraith is how his model can be used to coordinate tasks. The following are some of his suggestions:

It is advisable in real life (uncertain) situations to try to specify and match goals at a relatively high level of abstraction such that all the details can be handled in isolation by separate persons or units. On the other hand, where it is not possible to skip the details, considerably more effort is needed to integrate the tasks.

In line with this, the first two strategies (slack and self-containment) suggest that the interdependencies between groups should be minimized in order to reduce the amount of coordination. This obviously suggests that interdependence between tasks and the amount and type of coordination are related. Remarkably, Galbraith pays very little attention to the definition or conceptualization of interdependencies between tasks.

This gives rise to some interesting questions:

- What do interactions between tasks consist of and how should self-contained groups be created?
- Do appropriate coordination mechanisms depend on the type of dependencies between tasks? Put differently, is it always possible to apply goal-setting, or do some interactions force management to go into more detail?

Thompson provides more insight into these questions, and will be discussed below.

### *3.1.3 Structuring organizations according to Thompson*

Thompson argued that there is no reason why all tasks within an organization should be equally interdependent and all require identical coordination effort. His work is well known for its categorization of three conceptual types of interdependencies within the technological core of the organization. These are the following (pp. 54-55):

- *Pooled*: two parts are pooled interdependently if each part renders a discrete contribution to the whole and each is supported by the whole. Two plants may perform adequately without each other, but failure of one may hamper the overall organization and thus the other unit. Mintzberg (1979) and De Leeuw (1997) interpret this type as members that share common resources, but are otherwise independent.
- *Serial*: two parts are serially interdependent if one part produces an output which becomes the input for another part. The one part must act properly before the other, and both have a direct interdependence. Just as in a relay race, the baton is passed from runner to runner (Mintzberg 1979).
- *Reciprocal*: two parts are reciprocally interdependent in a situation in which the output of each becomes the input for the other. Thompson illustrates this by an example of an aircraft company's operations and maintenance units. A maintained plane is input for operations, and a plane back from its flight becomes the input for maintenance. Some authors interpret this interaction type as a pattern where each units inputs are its own outputs, recycled through other units (Victor & Blackburn, 1987).

According to Thompson, the contingencies of the types of interdependencies are additive. Reciprocal interactions by definition also include serial interactions that in turn also contain pooled coupling. Basically, the type or amount of coordination critically depends on the constraints (conditions) that are included by the interaction, as Thompson argues:

> *In the order introduced, the three types of interdependencies are increasingly difficult to coordinate because of increasing degrees of contingencies. (p.55)*

This results in the following claims:

- *Pooled coupling fits coordination by standardization*: This involves the establishment of routines, procedures or rules to constrain action consistent with those of others. Reliance on rules is most appropriate when there are few rules and these refer to relatively stable and repetitive situations.
- *Sequential coupling fits coordination by plan*: This includes the establishment of schedules for the interacting units. The amount and frequency of coordination is much higher, and stability and routine is far less necessary than in the first situation.

- *Reciprocal coupling fits coordination by mutual adjustment*: With this, during the performance of tasks, new information about each other's activities is processed. The flow of information has an iterative character. The more variable and unpredictable the situation, the greater the reliance on mutual adjustment.

With this in mind, Thompson states that organizations should structure their tasks such that overall coordination effort is minimized. Tasks must be clustered such that the strongest interactions occur within the groups, and the weaker interactions between the groups. Ideally, this would enable the reciprocal interdependent tasks to be mutually adjusted within a team, and planning and standardization can establish interactions with the remaining organizational parts. The coordination between the teams is referred to as system-level coordination.

Since groups have a limited optimal size and capacity, an organization must be structured in a hierarchy of groups if there are too many strongly coupled tasks for one group. Thompson states:

> *Those with the greatest interdependence form a group, and then the resulting groups are then clustered into overarching second order group(s).*

As a result, the hierarchically higher groups handle those aspects of coordination that are beyond the scope of any of its components. It will be noted that this is similar to Galbraith's model, though Galbraith goes one step further by adding lateral roles and information systems in order to broaden the scope of each of the units and to reduce the workload of the hierarchically higher units.

Remarkably, while Galbraith pays very little attention to the concept of interdependencies, Thompson does not explicitly include uncertainty. That is to say, Thompson locates the interdependent tasks within the company's technical core and hence technical and environmental uncertainty is reduced by the 'outer' layer. Each of the tasks are therefore performed under conditions as close to certainty as possible (Mintzberg 1979). However, uncertainty plays an important role within the three types of interdependencies. Terms previously used (e.g. hampering, stable, routine, and unpredictable) in the discussion of how interactions are linked to coordination mechanisms would suggest that in fact, Thompson and Galbraith have a similar way of reasoning. Thompson argued that the differences in interdependencies are based on the number of conditions that are included in an interaction. Logically, the more conditions involved within an interaction, the more likely it is that one of the conditions will not be met, and the more frequent and intense coordination becomes. This issue will be further explored in section 3.1.5.

To sum up, Thompson and Galbraith advocated the construction of semi-autonomous teams. However, in the next section attention will be paid to alternative considerations when structuring organizations.

### *3.1.4 Alternative structures of organization*

In addition to minimization of coordination of the information flows between the tasks, there are many other aspects that may play a role when structuring organizations.

Numerous studies have been conducted into effective organizational design. Proposed structures range from clustering tasks based on specialization, client groups, geographic location, or product mix, to work flows. Mintzberg isolated four basic criteria for structuring organizations. In addition to workflow interdependencies (similar to Thompson), these are process interdependencies, social interdependencies, and considerations of scale.

Process interdependencies relate to exchanges of information between specialists not necessarily working on the same work flow. For instance, electrical engineers have to consult each other about applications of the newest battery technology. Grouping of specialists encourages learning and in-depth knowledge of specific specialized fields.

Social interdependencies relate to the 'getting along' that accompanies daily work. To some extent, effective social relationships improve individual task performance and facilitate communication and working together. According to Galbraith, social relationships prosper best between members of the same discipline. Thompson, however, claims that the technical structure provides the major back-up for social structures.

Considerations of scale are related to the efficient use of capacity. For instance, a central university library is probably more efficient than a collection of faculty libraries due to economies of scale. The construction of semi-autonomous groups has a negative influence on scale efficiency, and vice versa.

To sum up, people can exchange information because their workflow is coordinated, or there is functional learning, or friendly communication (Kratzer et al. 2001). The structure of the organization has to take these three things into account, and also address scale aspects. Changing a structure towards optimization of workflow coordination is generally at the expense of functional learning. Simply put, structures are a matter of priority. Hence, stable firms with low time pressure and high focus on scale and expertise profit from a more functional way of structuring. On the other hand, companies competing on time-to-market and short cycle times would strongly benefit from an arrangement according to Thompson and Galbraith.

### *3.1.5 Discussion*

According to the classic paradigm, interdependencies can be identified on the basis of the critical contingencies facing the organization (tasks, technology, and environment) and an organizational structure that fits this set imposed. The central issue here is how all tasks can be coordinated in an efficient manner. Both Galbraith and Thompson advocate the construction of self-contained teams and introduce coordination mechanisms that handle the remaining system-level interactions.

The beauty of Thompson's work is that it distinguishes the reason for coordination (the interactions between the tasks) from the coordination activities themselves. Based on the characteristics of the system that has to be managed, one is able to suggest appropriate coordination devices to manage the interactions. This is an attractive idea and important for this thesis. The logic is simple and effective. The conditions that are included in an interaction determine the adequacy of a coordination mechanism. The more conditions are included, the more intense coordination becomes.

However, when it comes to the exact definition of the interaction constructs and their link with coordination mechanisms, some major issues arise. Several authors have criticized the applied constructs. Mintzberg, for instance, mentions the difficulties researchers have in applying the constructs in real-life complex situations. Victor and Blackburn (1987) challenge their practical utility since it is impossible to find out how much weaker a pooled interdependence is than a serial one. How can three pooled interactions be compared with one serial one?

Thompson's work also requires considerable investment of energy to understand its ramifications. Questions that need to be raised include the following: is pooled coupling always weaker than serial coupling, what is the exact difference between serial and reciprocal, and how should a chain of multiple tasks and interdependencies be handled? These will be examined below. These questions should be seen as a starting point for further discussion of these constructs. It is not within the scope of this research (however tempting it may be) to formalize the definitions and structurally examine all the potential flaws. After these considerations have been put forward, and interpretation valid for the remainder of this research will be settled on.

The first question is whether a pooled interdependence is always weaker than a serial one. This research would suggest that the answer is negative. If it is assumed that interdependencies of a pooled nature refer to the sharing of the same resource, this type may require a major coordination effort. In fact, when a resource is scarce and the organization is striving for optimal efficient use of the resource, coordination may become very sensitive to small exceptions and changes. One can imagine that detailed planning, or even mutual adjustment, may be required in these situations. According to the above reasoning, standardization would also add to the costs involved. It is very likely that a standard rule is not as effective as a detailed plan or mutual adjustment when optimal usage of a scare resource is involved. Hence, it is perhaps better to argue that while pooled interdependence corresponds minimally to coordination by standardization, there may be many alternative ways of handling this interaction that require more intense coordination mechanisms. Moreover, a serial includes more conditions (sequence) than pooled interdependence and these require minimum coordination at the very least. Standardization will not do for a serial, but that is not to say that mutual adjustment is not appropriate for serial interdependence.

Hence, it is suggested that the *minimum* coordination effort that goes with a pooled interdependence is lower than that for a serial. The conditions that are included within an interaction determine the *minimum* amount of coordination that is required, though there may be many good additional reasons why a pooled interaction requires more coordination than a serial one.

The second aspect has to do with the definition of reciprocal interdependence. The example given for reciprocal interdependency is in fact an example of two serial interdependencies, one of operations in relation to maintenance, and one of maintenance in relation to operations. Why is mutual adjustment suddenly seen as being necessary for two sequential relationships where planning was sufficient for one? The most likely explanation is that Thompson's example of reciprocal interdependencies is not correct. However, what is reciprocal interdependence? The only thing that can be suggested is that a reciprocal relationship refers to an unsolved problem similar to the previously described coupled design parameters within axiomatic design.

Problems with interdependencies become more serious if multiple interactions are considered separately. One can also question what exactly is meant by pooled interdependence in the original definition by Thompson. This is not the place to pursue these questions. It can, however, be concluded that Thompson's work provides fundamental insights, though the theory of interdependencies needs major updating if it is to be of more utility. Thompson's constructs will thus not be literally applied, nor its coordination mechanisms. Instead, what can be learned from the previous sections will be formulated and applied during the remainder of this research.

### *3.1.6 Lessons*

The next chapter will illustrate what recent approaches in product development have in common with the classic perspective. The following issues should thus be kept in mind during the remaining chapters.

- Interaction between tasks by definition requires coordination effort.
- Innovative companies have to deal with a great number of specialized tasks and consequently coordination is essential for performance.
- For these companies it is of crucial importance to find organizational structures that facilitate coordination effort. To that end theory advises that for the purpose of speed and efficiency the companies should arrange their work into semi-autonomous tasks.
- Coordination between these semi-autonomous groups of tasks is referred to as system-level coordination.
- Goal-setting is an efficient means of achieving coordination across tasks. Matching goals are specified at a relatively high level of abstraction and the details can be performed in isolation by separate units or individuals.
- When it is not possible to apply goal-setting, considerably more effort is needed to integrate the tasks, i.e. mutual adjustment.
- In order to understand the coordination problem it is essential to clearly distinguish what needs to be coordinated from how coordination is achieved.
- All that can be said about appropriate coordination mechanisms is that the characteristics of coordination will depend on the conditions that are included in an interaction. The more contingencies an interaction contains, the more effort is needed to integrate the tasks. It is more difficult to find appropriate conditions for the tasks, and the more conditions are included the higher the probability that some conditions will not be met.

Note that the above remarks on Galbraith's theory have been combined with Thompson's insights. The following chapters will refer to characteristics of coordination as the extent to which goal-setting can be applied.

## 3.2 Organization of design project

So far, the perspective of interdependencies between tasks has been described and it has been pointed out that the structuring of tasks is an essential variable for effective and successful organization. In the present section, recent literature that shows that a perceptive understanding of design team task structures offers a promising way of improving the performance of design teams will be discussed.

An explicit view of task structures is a relatively new concept within innovative environments due to its reputedly unique and unpredictable character. The specific characteristics of innovative environments will thus need to be described. Four different types of innovation projects will now be described, and it will be argued that these represent 'incremental' types of innovation. There will then be a brief exploration of the characteristics of design teams and common organizational practices, and this thesis's particular focus delineated. Design Structure Matrix (DSM) studies will then be explored, in particular how they are used to understand and improve system-level coordination during a design process. These studies are an essential starting point for this research and their strengths and weaknesses will be described in depth. The following aspects will be considered:

- Innovation projects: their scope.
- Project teams and their characteristics.
- DSM studies.
- Discussion of the DSM approach.

### *3.2.1 Types of innovation: scope*

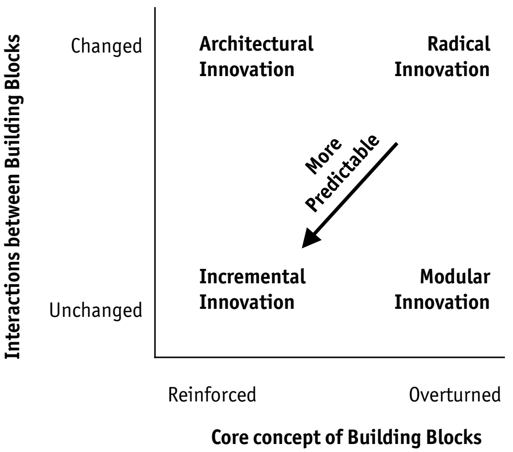
In general, innovation is associated with creativity, lack of routine, uniqueness, long time-span, and 'organic' structures. While it is logical to speak of structures in manufacturing, it seems almost paradoxical to analyze structures within the context of product development. Is it realistic to analyze structures in design processes, and if so, in which particular situations? In which situations it makes sense to analyze design project structures will be set out below.

Finding answers to the above questions is only possible if it is realized that not all design projects are equally innovative. Some involve the introduction of a radical new concept and others deal with refinement of an existing design. Henderson and Clark (1990) have identified four major types of product innovation based on two aspects. The first addresses the newness of the interactions between the building blocks of the product to be designed. The second relates to the newness of the core concepts of the building blocks. As shown in Figure 3.1, this results in four types of innovation: incremental, modular, architectural, and radical innovations.

Radical innovations implement new interactions and new product core concepts and are the most unique. This type of project involves new and uncertain operations and attacks the usefulness of established organizational working structures. Henderson and Clark describe the task faced by the project team as the generation of a new and appropriate working structure by trial and error during the operation. Hence, tasks and task structures are very difficult to analyze where there is radical innovation since these are subject to constant changes. However, if one moves to the bottom left corner of the Figure, the underlying structures of the design projects become more clear and stable. The more innovations can be characterized as incremental the more it makes sense to talk about more or less efficient working structures since the identified structures remain valid for a relatively long time (Eppinger et al. 1994, Gulati & Eppinger 1996, Smith & Eppinger 1997a, Smith & Eppinger

1998, Smith & Morrow 1999). Nowadays, innovation projects are viewed as having virtually the same merit as incremental ones. Due to speed and efficiency demands, companies are deliberately applying platform strategies (Meyer & Utterback 1993, Meyer et al. 1997) with more frequent but smaller increments of innovation. The success story of the Sony Walkman is a prime example in this context (Sanderson & Uzumeri 1995). Furthermore, by nature, organizations tend to embrace existing products as a basis for new innovations. In Chapter 4 this will be dealt with in depth.

It should be noted that the categorization presented here is mainly applicable to modular architectures since for integral architectures both dimensions of the classification seem strongly coupled. Nevertheless, the distinctions are important because they provide some direction in deciding to what extent it is useful to analyze the task structures of design processes. In the following, the modeling of design tasks will be addressed further and placed in the context of incremental-like innovations.




► **Figure 3.1** Different types of innovation according to Henderson and Clark

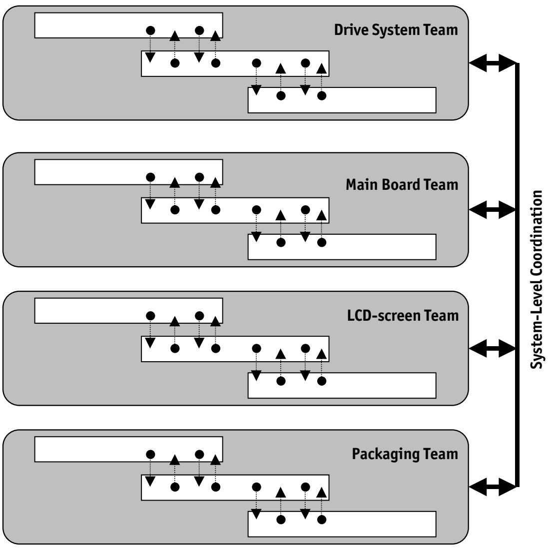
### *3.2.2 Design project teams*

The object of this study is the organizational structure of design project teams. These teams need to handle a large set of interrelated design tasks in order to achieve their overall goal – a new product. A brief description of design teams in general will now be given and the reasons for system-level coordination will form the focus of further attention.

As speed and functional integration become essential, many successful product development organizations have begun to work with project teams shaped according to a matrix structure. Matrix organizations combine the features of functional and process (project) organization. Organizational members are grouped according to the project they work on, and the functional area they represent. Consequently, each project member has two supervisors: the functional manager, and the project leader. This has recently become popular because it emphasizes operational flow and gives project management more priority than functional departments. These so-called heavy-weight project organizations have resulted in successful performance in the automobile and electronics industries (Wheelwright & Clark 1992, Baldwin & Clark 1997, Ulrich & Eppinger 2000).

Inspired by the Concurrent Engineering approach, communication across the members of a design team is facilitated to a considerable degree. Designers work in close collaboration, phases are overlapped, much of the coordination is lateral (Galbraith), and several engineering tools are in common use. These tools include Design for Manufacturing, Quality Function Deployment, and Failure Mode and Affect Analysis, and they stress the importance of and guide the integration of many design decisions, though obviously it is up to the engineers themselves to carry out the integration activities.

The size of a project team depends critically on the size and complexity of the product to be designed. For products such as automobiles, laptops, and electric shavers, a considerable number of engineers are required to get the job done. Large overall project teams are generally split up into smaller design teams that are each responsible for a part of the design work. In these cases it is not only important to coordinate all the work within a design team, it is also essential to compensate for the fact that these sub-teams behave as though they are fully independent. The interactions between the design teams have to be effectively managed to achieve the right overall design. According to classic organization theory, this is defined as system-level coordination. McCord and Eppinger (1993) refer to this integration problem as Concurrent Engineering in the large, and is illustrated in Figure 3.2. The figure depicts a laptop design project that is divided over four design teams. The designers within the teams need to put a great deal of effort into overlapping stages and integrating activities (i.e. Concurrent Engineering in the small), and between the teams particular attention is required to integrate them into a whole.




▶ **Figure 3.2** Concurrent Engineering in the large for a laptop based on McCord and Eppinger 1993

Attention will now be directed at an analysis of system-level coordination within large design teams. As will be shown later on, system-level interactions are often poorly understood within design teams but have a crucial impact on project performance. It will be argued that increased understanding of system-level coordination within product development is an excellent way of improving project performance based on available organizational theories.

In section 3.1 it was mentioned that classic organizational theories argued that 1) efficient organizational structures that minimize the need for system-level coordination are available, and 2) appropriate coordination mechanisms have to be applied to manage interactions between tasks.

These insights will be applied following an overview of tasks and related interactions within the design project. The DSM approach that provides a means of clearly depicting structures in design will now follow, and based on this, options to improve the design process will be generated.

It should be noted that from this point on, an organization will be considered a *design project*. Accordingly, the term ‘organizational unit’ (used in the classic theories) will refer to a *design team* that is a part of the *design project team*.

### *3.2.3 Design Structure Matrix studies*

Design Structure Matrix (DSM) studies highlight the fact that decomposition of a design project into smaller tasks is a variable of considerable managerial importance within product development and in particular for large and complex design processes. They suggest that deliberate coordination of interacting tasks is an issue of paramount importance and generally form the basis for improved project performance. DSM advocates firmly believe that a comprehensive and precise understanding of the flows of information between the tasks of a design project is essential. They provide detailed models showing why and where coordination is required, evaluate the efficiency of the project as a whole, and generate advice that can be used to improve future design projects.

The concept of DSM was introduced relatively anonymously in the early eighties by Steward (1981). After Von Hippel (1990) suggested that partitioning the design process is a decision that will heavily impact on design process efficiency, the idea attracted more attention. In the mid to late nineteen-nineties, the DSM studies by Smith and Eppinger (Eppinger et al. 1994, Smith & Eppinger 1997a, Smith & Eppinger 1997b, Smith & Eppinger 1998) forced a breakthrough in management science. Recently, the DSM homepage claimed that the models have been applied to many issues and it is expected that they will gain even more popularity in the coming years. One simple reason for the high recognition is perhaps that many academic researchers in operations management have shifted their attention towards product development and they frequently prefer the DSM models because of their suitability for quantitative modeling and optimization. A more fundamental and qualitative reason is that the DSM approach gives some substance to conceptual organizational theories, which makes it an attractive instrument for real-life cases in general. This is the line that will be followed here when considering the DSM approach.

The major advantage of DSM is that it provides an overview of the structure of design tasks. How DSM studies in general collect their data and model these in a matrix will be described, as well as the range of applications for which design structure matrices are suitable. Studies that address the decomposition and integration decisions of large and complex design processes will be focused on. The features of this approach will be summarized first, and its benefits, limitations and criticism described. The ultimate aim is to arrive at an approach that relies on DSM’s strengths. The reasons for rejecting some elements will be expanded on in the ensuing chapters.

*Documenting tasks and interactions*
DSM studies model design tasks to be performed during a design process, and document the required flows of information between the tasks. This data is collected retrospectively by interviewing project members and management. The first step in the approach is documenting all of the tasks that need to be performed by the project members. In the next step, the interviewees are asked the following questions: 'Which task(s) provide information necessary to perform your task?', 'How often do you need to get technical information from other tasks in order to complete your task?'. Tasks are thus modeled as transformations of information, and interactions are defined as exchanges of information between the tasks (in the perception of the interviewees).

Next, the data are captured within a matrix. The tasks to be performed are arranged in identically labeled rows and columns in the matrix. In turn, the DSM elements represent the interactions (input and output of information) between each possible pair of building blocks. More precisely, element (i,j) indicates whether task i requires the output of task j in order to be performed. Figure 3.3 illustrates 9 design tasks in an arbitrary design process. The task 4 column indicates that tasks 5, 7, and 9 need input from 4 as a condition to be performed. Consequently, interdependence between each task pair (a,b) can be characterized as:

- *Independent*: Neither task is involved in an interaction. Note that pooled dependencies are not included here.
- *Dependent*: Task a requires the output of task b (or vice versa). This corresponds to Thompson's sequential interdependence.
- *Interdependent*: Task a needs information from b, and b requires the output of a, comparable to Thompson's reciprocal interdependence.

| | Task 1 | Task 2 | Task 3 | Task 4 | Task 5 | Task 6 | Task 7 | Task 8 | Task 9 | |
|---|---|---|---|---|---|---|---|---|---|---|
| Task 1 | **x** | | | | | | | | | |
| Task 2 | | **x** | | | | | | | | ► 'Independent' |
| Task 3 | | **1** | **x** | | | | | | | ► 'Dependent' |
| Task 4 | | | 1 | **x** | **1** | | | | | ► 'Interdependent' |
| Task 5 | | 1 | 1 | 1 | **x** | 1 | | | | |
| Task 6 | 1 | | 1 | | 1 | **x** | 1 | | | |
| Task 7 | | | 1 | 1 | | 1 | **x** | 1 | 1 | |
| Task 8 | | 1 | | | | 1 | 1 | **x** | 1 | |
| Task 9 | | | 1 | 1 | 1 | 1 | 1 | 1 | **x** | |

► **Figure 3.3** The Design Structure Matrix

*Applications*
In general, DSM applications can be arranged into two categories (Smith & Morrow 1999). The first category is that of the scheduling of design tasks and the identification of iteration in design; the second is that of the decomposition and integration of large design projects.

*Scheduling and iteration*
The first category will be very briefly described. The models analyze the throughput times of design processes by identifying reciprocal interactions, and estimating the time that is needed to execute the corresponding iterating design tasks. Smith and Eppinger modeled the

design tasks for an automotive brake system and concluded that an analysis of design iterations was able to provide an accurate estimation of duration and potential variation in lead time. Alongside this, familiar (or extended) models focus on concurrency, moment of overlap, costs, clustering and scheduling of strongly coupled tasks. Smith and Morrow (1999) and Krishnan and Eppinger (Krishnan et al. 1997) demonstrate that concurrent execution of tasks in order to speed up the design may have contradictory results. They showed that too much overlap of tasks, even those that by nature are sequentially related, may cause repetition of work and slow down the design process. These studies generally suggest two means for improvement – reduction in the number of iterations, or speeding up the iterations. The second category of applications is that of the decomposition and integration of large design projects and will now be described and discussed.

*Decomposition and integration of large design projects*

A few studies have applied DSM models to analyze how a project team is decomposed into sub teams and how their system-level interactions are managed. McCord and Eppinger (1993) modeled a situation involving the design of a laptop (see Figure 3.2) and created the DSM shown in Figure 3.4. They identified the design teams and their main design tasks and documented the corresponding exchanges of information. All of the flows are illustrated in a DSM and provide an overview of the required information transfers across the whole project. For a design project involving a small block V8 automotive engine, such a representation provided a sound basis for improvement of the design process.

First, project team understanding of their system-level coordination needs was increased. Management was made structurally aware of the importance of the decomposition decision for the project as a whole. Prior to the analysis, their understanding was based on intuition and past experience alone. Furthermore, engineers started to take more conscious notice of how their work impacted on the overall process, or how their work could be affected by other tasks.

Second, the representation resulted in structural 'Thompson-like' recommendations. The process could be improved if:

- Teams were regrouped around the most dependent tasks such that these could be adjusted mutually.
- Appropriate coordination mechanisms to handle the remaining system-level interactions were suggested. Formal coordination modes are proposed as an alternative for relying completely on informal coordination and expecting designers to continuously supervise the process. McCord and Eppinger refer in this context to task forces, liaison roles, management hierarchy, and so on.

Alternatively, Eppinger et al. (1994) have advocated manipulating the existing set of interactions in order to 'delete' some system-level interactions and to reduce the need for coordination. All of the DSM advocates argue that these suggestions will have a strong positive effect on the quality and speed of the design project. Quality will improve since the sub-problems will be better integrated, and development time will decrease since interactions are better managed and parallel processing is facilitated.

It is also important to note that the suggestions are not actual improvements. They in fact show how an already finished project could have been managed better. They must be interpreted as valuable lessons learned that can be implemented in similar incremental design projects in the future.

| | Drive System Team | | | | | Main Board Team | | | | | LCD Screen Team | | | | | Packaging Team | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Drive System Team | x | 1 | 1 | | | | | | | | | 1 | | | 1 | | 1 | | | |
| | 1 | x | | | 1 | 1 | | 1 | | 1 | | | | | | | | | | |
| | 1 | 1 | x | 1 | | 1 | | | | | | | | 1 | | | 1 | | | |
| | 1 | 1 | 1 | x | | | | | | 1 | | 1 | | | | | | | | |
| | | 1 | 1 | 1 | x | 1 | 1 | | | | | | | | | | | 1 | | |
| Main Board Team | 1 | | 1 | | 1 | x | 1 | | | | | | | | | | | | | |
| | | | 1 | 1 | | 1 | x | | 1 | | | 1 | | | | | | | | |
| | | 1 | | | | 1 | 1 | x | | 1 | | | | 1 | | 1 | | | | |
| | | | | | | 1 | | 1 | x | 1 | | | 1 | | | | 1 | | | |
| | | 1 | | 1 | 1 | 1 | 1 | 1 | 1 | x | | | | | | | | | | |
| LCD Screen Team | | | | | | | | | | | x | 1 | | | | | 1 | | | |
| | | | | | | | 1 | | 1 | | 1 | x | | 1 | | | | | | |
| | | | | | | | | 1 | | | 1 | 1 | x | 1 | | | 1 | | | |
| | | | | | | | 1 | | 1 | | 1 | 1 | | x | 1 | | | | | |
| | | | | | | | | | | | | | 1 | 1 | x | | | | | |
| Packaging Team | 1 | | | | | | | | | | | | | | | x | 1 | 1 | 1 | |
| | | 1 | 1 | | | | 1 | 1 | | | | | | | | 1 | x | 1 | 1 | |
| | | | | 1 | | | | | | | | | 1 | | | 1 | 1 | x | | 1 |
| | | | | | | | | | | | | | | | | | 1 | 1 | x | 1 |
| | | | | | | | | | 1 | 1 | | 1 | | | | 1 | 1 | 1 | 1 | x |

▶ **Figure 3.4** DSM of system-level communication between design teams (adapted from McCord and Eppinger)

### *3.2.4 Discussion of the DSM approach: strengths and weaknesses*

This section will discuss the DSM philosophy and focus on its ability to model and improve system-level coordination. It will start with a summing up of its advantages and end with what could be regarded as some serious limitations.

- The strength of DSM is its clear overview of tasks and interactions and its suitability for a wide range of structural analyses.
- The approach links the technical transfer of information to organizational consequences, where this relationship is often ignored. It highlights the crucial issue of decomposition that is generally poorly understood, and managed on intuition or previous experiences.
- The general understanding and consensus about all flows within a project team is an important aspect that, in addition to organizational theories, will in itself trigger improvement.
- The clear representation provides an incentive for applying (classic) organizational principles at a highly subtle and detailed level such that it has high practical and theoretical relevance. The organizational theories themselves provide valuable insights, but are too conceptual to be of direct support in practical situations.
- The DSM philosophy is based on the idea that options for improvement always need to be placed within the actual context of the problem. This is an essential element, and more realistic than prescribing 'optimal' ways of working without detailed understanding of the current situation.

Despite the undoubtedly strong features of DSM, it also has some serious drawbacks. The validity of the interactions measured is open to doubt, and it is hard to see how the current analyses are able to actually support and direct future improvements.

In the following, three characteristics that a DSM analysis should ideally possess to be valid will be noted; these will be of use for future design projects. There will then be a critical review of DSM research to establish whether the criteria can be met.
First, the following criteria can be identified:

- The interactions that are measured should be valid. Each documented interaction should be generally understandable and non-subjective.
- Ideally, it should be possible to deduce which specific characteristics for coordination match each documented interaction. This will help provide a clear explanation of the coordination efforts in a 'current' design project, and help to suggest appropriate coordination devices.
- In order to actually implement improvements one obviously needs to know how an interaction can be manipulated (by what decisions), and to what extent an interaction structure can be changed, if at all.

Second, a major drawback in current DSM research is that exchange of information is based on a much too general definition of interaction to be able to improve practical situations.

- The construct 'exchange of information' is very broadly interpretable, and it is difficult to know whether everybody is talking about the same interactions. There may be many factors that may cause a (perceived) exchange of information. Since the underling cause of an interaction is unclear, the gathering of interaction data may be affected by the focus, background, perception of the interviewees and the moment of interviewing (Staudenmayer 1999, Oosterman ). McCord and Eppinger, for instance, claim that different interviewees had different opinions about information exchange between the same tasks. Moreover, documented interaction patterns may not only refer to work flows, but also to functional learning or friendly communication (as described in 2.1.). Despite most studies clearly stating that they refer to work flows, it seems very likely that the various types have been mixed up by the interviewees. To sum up, the validity of the measured interactions is thus questionable.
- DSM models do not clearly distinguish what needs to be coordinated from the coordination activity itself (how do we coordinate?). This distinction is needed to understand the reason for coordination and to be able to deduce the specific characteristics for coordination (see the discussion of Thompson's work in section 2.1). However, DSM models model coordination activities as information transfers between tasks or teams, so it is difficult to distinguish between interactions and coordination within the current conceptualization of interactions. This raises many questions. Does an interaction imply the need for coordination, or is it the result of a particular coordination activity? Which interactions are embedded in the underlying structure of the design problem, and which are caused by the specific way that the project team works? What happens when an interaction is 'deleted'? Is it a coordination activity being deleted, or a need for coordination? Despite DSM studies seeming to first model interactions and then propose coordination mechanisms, these features are in fact hard to distinguish, which hampers validity and interpretation.
- In line with the previous remark, it is not possible to define which conditions are included in an interaction (they all refer to information). As a result, the specific characteristics (i.e. level of detail) of coordination mechanisms cannot be deduced (see the discussion about Thompson). Furthermore, it is completely unclear how (by means of what decisions) an

interaction can be manipulated, or to what extent an interaction structure can be changed, if at all (given the company's policy). Moreover, given the vague definition, it even is questionable whether a specific interaction will ever return in a future set.

### *3.2.5 Summary*

In Chapter 1, this chapter's objective was formulated as being to examine how management science represents system-level coordination within design processes, and which classic organizational principles are available to improve system-level coordination.

Accordingly, it has been shown that classic organization theory states that within product development situations it is advisable to group the design work within semi-autonomous groups of design tasks that can each be handled by a separate organizational unit (design team). As a logical consequence, great emphasis is placed on system-level coordination between the organizational units. A hierarchically higher unit should handle those issues that are beyond the scope of each of the single units, and lateral exchange of information between the units is advocated.

Inspired by Galbraith, it has been suggested in particular that the approach of goal-setting is an effective means to achieve coordination. The goals of two units have been specified at a relatively high level of abstraction, such that the detailed actions can be performed in relative isolation by each team separately. If for some reason it is not possible to apply goal-setting, intense coordination effort between the teams is required. Thompson argued that in order to indicate which coordination mechanisms are appropriate, the characteristics of the interactions between the tasks have to be considered. The more conditions included in an interaction, the more difficult and intense the corresponding coordination becomes. Hence knowledge of the characteristics of the system that has to be coordinated provides information about which coordination devices can be applied.

The second section illustrated the DSM approach that documents exchanges of information between design teams. Based on a matrix representation that depicts the design teams and their exchanges of information, a clear overview of the system-level coordination problem was provided. This detailed modeling forms the basis for detailed understanding of where system-level coordination is required between teams, and provides an incentive for generating options for improvement. Constructing more independent groups of tasks (in order to reduce the need for coordination), or applying appropriate coordination mechanisms to handle the interactions are some of the options suggested. They must be considered lessons learned that will improve the performance of future design processes. In the discussion, however, it was concluded that these models only have a limited capacity to make a proper analysis. The way the interactions between teams are modeled (exchange of information) is much too broad to achieve a clear understanding or to suggest clear options for improvement. This means that the above-mentioned classic theories are of little practical use.

The next chapter will examine the issue of product architecture and look for ways to enrich the DSM way of modeling.

# 4 Coupling Product Architecture and Organization

Although product architecture and design task structure are often considered separately, they have a considerable impact on each other. In this chapter, the architectural and organizational pieces of knowledge will be brought together and suggestions made for pieces that need to be added such that both scholars and practitioners will eventually benefit from the coupling of expertise relating to the two areas.

Since architectural knowledge available in engineering design was first described, followed by the organizational aspect of design processes, architecture and organization will be linked in the same order. How a particular architecture affects system-level coordination during a design process will be explored. It will be argued that comprehensive knowledge of a particular product architecture has the potential to be able to explain system-level coordination and to identify effective measures for improvement (in line with the prior discussion of the DSM approach).

It should, however, be noted that this is not the only way to explore this relationship. Alternatively, product architecture may also be analyzed as being the result of an established organizational structure. In fact organization and architecture are strongly interrelated, despite the fact that the usual focus is on how architecture affects the organization and not the other way around. The following approach will be taken.

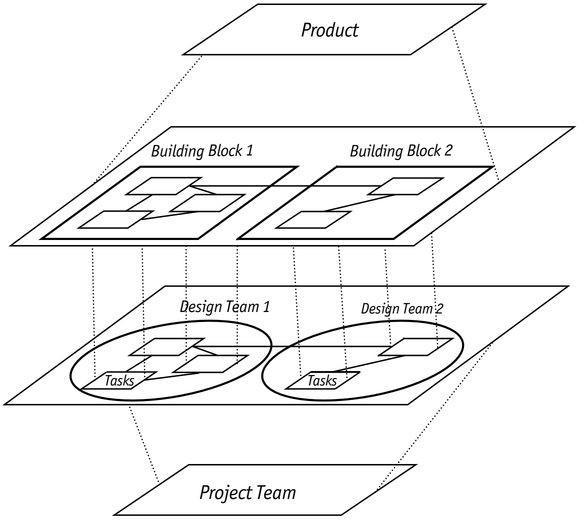
It will first be argued that effective companies describe their design work in terms of the architecture of a product, an important premise for the rest of the research. Second, general research into the relationship between architecture and organization will be explored. Third, how the DSM models of Chapter 4 may be strengthened by a detailed representation of interactions between product building blocks (Chapter 2) will be described. This will be done using a description and discussion of the study by Pimmler and Eppinger (1994). Fourth, the other side of the coin (how organization affects architecture) will be dealt with to ensure that the parallels are correct. Finally, the above research will be discussed and reviewed in the light of what has emerged from the previous chapters. The approach to be taken in the following chapters will be outlined, and the research goals relating to these chapters formulated under the following headings:

- Organizing tasks around building blocks.
- The consequences of architecture for coordination.
- Detailed analysis of architecture and system-level coordination.
- The consequences of organization for architecture.
- Formulating the problem.

## 4.1 Organizing tasks around building blocks

Product development processes and the problem-solving structures associated with them correspond to a large degree to the structure of the product to be designed (Simon 1981, Von

Hippel 1990, Wheelwright & Clark 1992, Gulati & Eppinger 1996). This relationship will be apparent if it is realized that the lion's share of tasks reflects the design of a piece of the product. Though the design tasks can still be organized into many possible forms, it can be argued that effective organizations in competitive environments group their design tasks around a product's building blocks. A combination of two prior concepts (often considered separately) makes this plausible. The first is that of interactions between the building blocks of a physical product. In Chapter 2 it was demonstrated that with all products, the interactions between a product's building blocks are generally weaker than the interactions within the building blocks. This characteristic becomes more dominant as the amount of modularity of a product increases. The second concept is that of the classic paradigm of effective organization. Chapter 3 showed that organizational theory advises structuring tasks into semi-autonomous groups of tasks in order to minimize overall coordination effort and increase speed. This is especially useful within competitive environments (Thompson 1967, Galbraith 1973, Mintzberg 1979). To sum up, effective design projects strive to allocate the conception of each building block to an organizational unit solving all interactions within the block. This is depicted in Figure 4.1. Novak and Eppinger (1998) noticed recently that effectively performing firms in the automotive industry have a good fit between organizational structure and the structure of the product. Moreover, several authors have claimed effectiveness for organizational structures that mirror the product architecture (Henderson & Clark 1990, Gulati & Eppinger 1996, Novak & Fine 1996, Sanchez 1999a).




▸ **Figure 4. 1** Reflecting product architecture within the organization

Assuming that the organizational structure mirrors the building blocks, the interactions between the blocks become of prime importance. The next sections will discuss how product architecture affects the organizational structure, and system-level coordination in particular. Research that takes place at a relatively high level of abstraction will first be discussed, and then the very few studies that link the two at a detailed level will be explored. The converse relationship (from organization to architecture) will also be addressed.

## 4.2 The consequences of architecture for coordination

Interactions between building blocks play a major role in understanding the need for system-level coordination during a product development project. Some deductions can be made based on the points of departure in the above section. The more modular the product, the more the building blocks can be designed in parallel, hence greater speed and greater self-containment of the organizational entities. Conversely, the more interactions between the building blocks of the physical product, the more system-level coordination is required to gear the groups of tasks to each other. The amount and type of interaction between building blocks thus affects the need for system-level coordination during the design process. Hence, detailed understanding of this relationship is of considerable managerial importance. Surprisingly, very few studies have elaborated this concept at sufficiently high levels of detail (Gulati & Eppinger 1996, Erixon 1998, Sosa et al. 2000).

Novak and Eppinger (1998) conclude that the type of product architecture will determine the choice of efficient organizational structures facilitating required system-level coordination effort. They found that building blocks of integral products can best be designed in-house in close cooperation with all designers since corresponding interactions require intense, product-specific, and frequent coordination. Alternatively, modular products need much less system-coordination and outsourcing the design of the building blocks is a feasible option. Companies that apply outsourcing strategies in combination with integral product architectures generally perform much more poorly than firms which match product and organizational structures.

Henderson and Clark (1990) illustrate the crucial role of communication channels between the organizational entities responsible for the design of a building block. An examination of architectural innovations in the photolithography alignment industry shows that coordination mechanisms have to closely match the characteristics of technical interactions between the building blocks. Cases in which coordination did not manage the interactions sufficiently well cause painful situations. The authors mention that there is little need to lay emphasis on interactions within the blocks since these were naturally managed within each unit and have less effect on the whole.

Staudenmayer (1999) expands what is known about particular coordination mechanisms in product development projects. She has shown that the type of product architecture will affect the intensity and type of coordination strategy. She analyzed a number of software design projects and categorized them into three architectural types: modular, hybrid, and integral.

For the modular cases, project members expended relatively high amounts of effort in making up-front architectural and organizational decisions. During the course of the project, there was high and standardized focus on specification of interfaces, strict ownership, and smooth day-to-day coordination. For the hybrid architectures, there was also a high focus on up-front decisions. However, these were restricted by broader contingencies preventing the choice of modular structures. During the design process some strong interactions between blocks were identified, documented and structurally communicated. Change protocols were strict and standard and frequent ad hoc coordination was required. The integral projects typically included few up-front considerations, yet involved a strategy of reacting to new situations during the design process without discussing overall effective structures. Interfaces were not standardized, interactions were solved informally, specifications were flexible and changing, and members felt they were spending too much time and energy in coordination and solving local conflicts. Overall there was tentative evidence that integral, locally responding projects performed more poorly (time, cost, quality) than the other two categories. Furthermore, the modular projects tended to be the best performers, but these findings were less readily interpretable.

Though situated in a software environment, these results suggest a match between architecture and specific devices in order to achieve coordination. Nevertheless, understanding of architecture and specific modes of coordination remains limited due to lack of clear illustration of the architectures of the elaborated products. The only feature that is described is an estimation of the amount of modularity of a product, without regard to the blocks or interactions. As a result the coordination activities are explored in great detail but the interactions are not. This hampers the finding of answers to more subtle but highly relevant questions: Are the differences between the coordination approaches a necessary consequence of the characteristics of specific interactions or not? Did the team members involved in the integral projects examined work in a less structured and effective fashion than is possible for the type of interactions, or is a reactive way of working inherent to the interaction structure? Furthermore, what are the reasons for a particular architecture, and what specific broader contingencies underlie these?

The above study thus gives rise to very interesting insights, but in order to obtain a deeper understanding (and to be of use for improvements) a more detailed representation of the particular product is needed.

To sum up, the studies described above reveal some important general principles:

- Technical interactions between building blocks call for system-level coordination.
- System-level coordination is a crucial aspect with respect to the performance of design projects. Interactions within design teams are much more naturally managed than the interactions between these.
- Different types of architectures match different coordination devices.

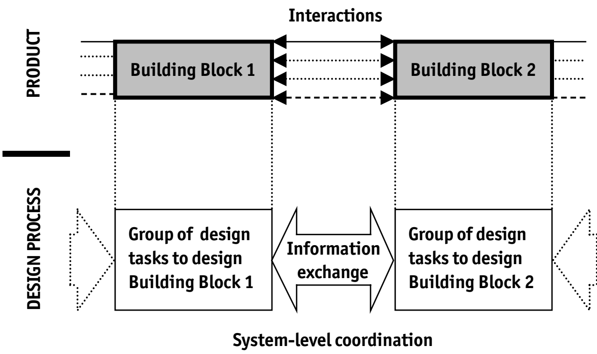
## 4.3 Detailed analysis of architecture and system-level coordination

It was noted above that the described studies analyze particular architectures at too high a level of abstraction to provide a basis for organizational improvement. For more detailed understanding, research available within engineering science and management science has to be sought. The problem, though, is that these are two separate streams and communication between them is rare. An exciting exception, however, is the work of Sosa and Eppinger (2000) who recently applied a taxonomy of interactions (see Chapter 2.3) to building blocks in an organizational context. Their approach has similarities with our research goals. Their work is thus a potential source of information and as such will inform the formulation of the problem at the end of this chapter.

### *4.3.1 Sosa and Eppinger: main idea*

Sosa and Eppinger link documented interactions between the building blocks of a product to the documented exchange of information between groups of designers. This is depicted in Figure 4.2. In fact, they combine of the two features that have been described in the foregoing chapters. The first is that of representing product architectures as an illustration of different types of interactions between physical building blocks, as described in Chapter 2. The taxonomy (after Pimmler & Eppinger, 1994) is comprised of energy, material, information, structural, and spatial types of interaction. The second facet includes the documentation of exchange of information between organizational units each facing the design of a building block during the design process. The overall patterns of information flow are structurally documented in a DSM as described in section 3.2.

For a single case study of a design project involving construction of a large aircraft engine and consisting of 54 components and 54 design teams, Sosa and Eppinger matched the interactions between 8 aggregate building blocks to system-level exchange of information between 8 corresponding groups of design teams.




► **Figure 4.2** Mapping the product on the design process

They found that in an average of 78% of the cases, an identified technical interaction (no matter what type, or number of interactions) between the blocks corresponds to actual system-level exchange of information during the course of the design process. The remaining 22% included cases where known interactions were not matched by system-level communication or cases where reported system-level communication did not correspond to documented interactions.

Without examining the detailed findings of this paper this would suggest that measuring interactions between building blocks has a high potential for clarifying the relationship between architecture and system-level coordination. Their approach will be discussed below.

### *4.3.2 Discussion*

Compared to the earlier discussion of DSM, this approach clearly distinguishes what needs to be coordinated (interactions between building blocks) from how coordination is achieved (the communication patterns between the design teams). For any product, it is easy to indicate where exactly interaction occurs and thus between which design teams coordination is required. On the other hand, however, the study offers limited understanding of specific coordination activities and gives limited insight into how future design processes can be improved. Despite in-depth technical knowledge of the product, the analysis does not result in a detailed understanding of interactions and system-level coordination. A number of remarks related to that concept can be made.

First, Pimmler and Eppinger do not translate the number of interactions between two blocks into differences in the amount of communication. The analysis only takes into account whether there is a zero or non-zero number of interactions. Two blocks that have an energy, material, and a spatial type of interaction are modeled in a similar fashion to two blocks that only have an energy type of interaction.

Second, it is left undecided whether a particular type of interaction by its very nature requires more or less coordination effort. Sosa and Eppinger (implicitly) consider each type as having similar consequences. Pimmler and Eppinger (the instigators of the taxonomy) suggested, however, that one type of interaction may have a different effect on coordination effort than another. However, they qualify this by saying that it is not possible to make a logical statement about whether (for instance) a spatial interaction requires more exchange of information than an exchange type.

Third, investigating whether each type of interaction imposes a specific coordination approach between design teams may produce interesting results. In Chapter 3 for instance, the goal-setting approach is introduced. These aspects have not yet been considered in this study, and as argued in the previous remark it is not possible to abstract such information from the current taxonomy.

Fourth, how the present analysis is able to suggest improvements to the design process under study (in line with the DSM philosophy) is a question that must be asked. This is difficult to find out based on current interaction constructs. For instance, with the spatial type of interactions it is very difficult to understand from where improvement can be expected to come and how the interactions can be manipulated (in order to reduce the need for system-level coordination). Is a spatial interaction a consequence of a specific (fixed) interface, a result of a side effect, or because two blocks together fulfil a function (and adjacency is required). This cannot be deduced, and this in turn hampers a thorough understanding.

Although Sosa and Eppinger probably did not aim to find answers to the above and it is probably impossible to find a solution for all the issues, the point is the centrality of the applied taxonomy of interactions. It is all about whether the types of interactions represent architecture in such a manner that it enables a useful analysis. This applies, in fact, to any kind of analysis. With this research, the kind of answers obtained regarding the consequences of organizational architecture were determined by the information included within each type of interaction. The papers by Sosa and Eppinger and by Pimmler and Eppinger pay remarkably little attention to the foundations of the taxonomy. As argued above, it is difficult to understand clearly what causes each interaction (especially spatial), how each type responds to coordination efforts, and how each type can be manipulated.

Hence, if one wants more than just an indication of whether exchange of information between teams is necessary or not, a close look needs to be taken at the interaction constructs that are being applied. In the formulation of the problem at the end of this chapter, it will be argued that measuring interactions between building blocks has a high potential for making the role of architecture within engineering science transparent, yet the need for a reconsidered taxonomy is also obvious.

In Chapter 5 a new taxonomy will be proposed, and the current taxonomy reexamined. Attention will also be paid to whether this taxonomy is sufficient for the purposes of analysis. Before doing so, some light must be shed on how organization effects architecture. The line this research will take will then be addressed.

## 4.4 The consequences of organization for architecture

In this section the relationship between architecture and organization will be viewed from the opposite direction. While this is not the main focus of this research it may serve to show to what end the representation of a particular product architecture is useful for future products. A more thorough explanation of the feature of incremental innovations as described earlier in section 3.2.1 will, in fact, be given.

What is known about how established organizational forms affect the structure of products is largely based on the work of Henderson and Clark. They build upon two concepts that are important for understanding this effect. The first is that technical evolution is generally characterized by a period of enormous experimentation followed by a particular design becoming dominant and accepted. As a result, the range of subsequent design projects takes the major decisions of building blocks and interactions as given, and the corresponding design teams have very similar task structures.

Their second argument is that organizations build their knowledge and structure around the recurring design tasks for each incremental innovation. In effect, the architectural knowledge tends to be implicitly embedded in communication filters and problem-solving organizational strategies. As tasks become more stable, organizations create filters that allow them to identify immediately the most relevant pieces of information from the enormous diversity of available information. For instance, communication channels between designers of a driving unit and a power unit will be shaped in such a way that they effectively handle the critical interactions and ignore all irrelevant information. The engineers of the driving unit are interested in the specifications of the energy supplied but do not need to know the color of the power supply. Organizations create information filters that reflect prior knowledge of interactions, and in this way deal efficiently with the enormous complexity of available information. Similarly, people familiar with the city of Groningen will probably recognize that years of experience of traveling by bike is no guarantee that you will find your way by car (without a fine for violated one-way signs). Traffic signs meant for cars do not apply to cyclists and are not noticed by them. This works effectively until new situations appear. In line with this, in the previously described study, Sosa and Eppinger found that system-level communication could be better predicted with 'integral' (relatively few internal interactions) blocks than with 'modular' (relatively few external interactions) blocks. They argued that designers of 'integral' blocks are far more used to managing incidental interactions with other blocks than designers of 'modular' blocks.

In fact, past products have strongly affected the organizational structure and habits of companies. In turn these experiences significantly impact on the design of future products. As Simon argues (Chapter 2.1) problem-solving strategies are shaped by previous experience that led to successful solutions. The established organizational structure with specialized tasks and filtered information will therefore greatly influence the structure of newly designed products. Organizations with a dominant design thus develop organizational boundaries, which are beneficial when similarly structured products undergo innovation, which in turn strengthens the established boundaries. This effect hampers more radical innovations to a considerable degree and stimulates the dominance of a particular architecture. Henderson and Clark showed that changing an architecture is extremely difficult and requires painstaking care. New interactions between blocks require new filters and implicit knowledge, but changing the corresponding capabilities of the firm is extremely time-consuming.

There is doubtless much more literature on communication filters, problem strategies and evolving organization. However, that mentioned above gives rise to the following two concepts:

- A thorough understanding of an existing product architecture is very likely to be useful for future derivative products.
- When it is proposed that an architecture be changed (i.e. to achieve organizational benefits) it is extremely important to understand what the established interactions look like, and not just to propose a new architecture.

## 4.5 Formulation of the problem

By this point it will have become clear that product development projects usually involve a lot of people whose work is characterized by frequent interactions. The management (coordination) of these interactions is of crucial importance for the effective performance of company design projects. There has been a particular focus on large project teams that are split up into smaller design teams and where there was a need to manage the remaining interactions between the teams. Chapters 3 and 4 examine a number of papers that had found that system-level coordination (coordination of interactions between design teams) is an essential variable for a project, and in many cases is a factor that can be significantly improved. The DSM approach (described in Chapter 3) showed that system-level coordination can be analyzed based on a subtle understanding of how the project team is decomposed into (interacting) design teams. A clear overall representation of the flows of information between design teams enables improvement of design projects in the following two ways:

- By applying appropriate coordination mechanisms to manage the existing interactions between the design teams.
- By reducing the number of interactions between the design teams in order to facilitate system-level coordination. This can be accomplished by a restructuring of existing tasks, or by manipulation of the interactions.

These ostensibly clear and simple principles seem attractive ways of understanding and improving design processes. However, in the final part of Chapter 3 it was argued that the current DSM models are only capable to a limited extent of making a correct analysis and useful options for improvement were suggested. The major point of criticism is that the interaction construct (exchange of information) is much too broad since it lacks a clear indication of the underlying causes for interaction. It is therefore difficult to deduce appropriate coordination mechanisms and to suggest how an interaction can be manipulated. Moreover, the sets of interactions are multi-interpretable and there is the serious danger of interaction and coordination activities being mixed up.

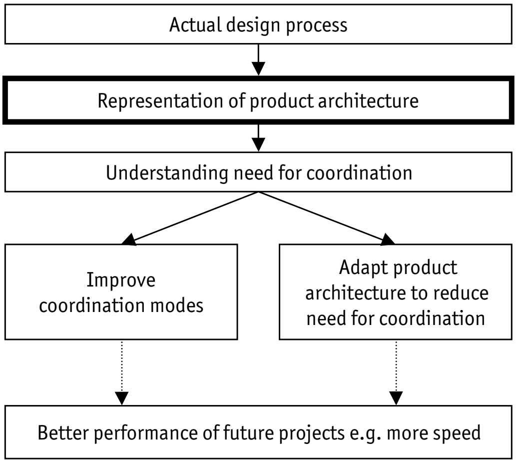
In this research the component of product architecture is added to that of design team structure. The reason behind this is that effective firms match their design project to the architecture of a product (see 4.1). This was illustrated using research highlighting this relationship (see 4.2). It will also be recalled that product architecture can be represented as a collection of physical building blocks that are involved in interactions of various types (see 2.3). Taking these concepts together, it could be deduced that when design teams mirror building blocks, the interactions between the blocks must logically correspond to the system-level coordination needs between the design teams.

It will now be proposed that instead of modeling exchanges of information between design tasks, interactions between building blocks should be considered and translated into consequences for coordination. This will produce a clear distinction between the reasons for coordination (the interactions between the building blocks) and the system-level coordination activities themselves.

The suggested approach to analyzing a design project is depicted in Figure 4.3. In order to analyze the system-level coordination of a design process, interactions between the building blocks of a designed product will be modeled. There will then be an examination (retrospectively) of the system-level coordination activities and these will be reviewed in the light of the identified interactions between the building blocks. The purpose of this is to show that the underlying interactions enhance understanding of the system-level coordination activities that take place within a project team. In turn, this understanding can be used to improve system-level coordination in two ways:

- By applying more appropriate coordination mechanisms to each type of interaction (this is shown in the figure as 'improve coordination modes').
- By manipulating the interactions between the building blocks such that coordination needs will be reduced (this is illustrated as 'adapt product architecture to reduce need for coordination').

The two suggested strategies can be applied simultaneously. The results they produce may also be valid as lessons learned and be useful for future design projects. Based on section 4.4, it may thus be reasoned that a subtle understanding and analysis of an existing product architecture provides very useful information for future derivative projects. Finally, it may be assumed that when the two options are applied to each interaction they will result in better performance of future design processes. In line with the DSM models, improved interaction structure results in more speed, higher quality and lower costs.




▸ **Figure 4.3** Diagrammatic illustration of the suggested approach

It was recognized that a particular product architecture potentially contains a lot of information useful for gaining an understanding of what is required for system-level coordination in a design process and for generating options for improving system-level coordination in order to increase project performance, including future performance. The question remains as to how to place the subtle types of information relating to product architecture within a useful conceptual framework. As was concluded in 3.4, it is all about which types of interactions can be identified so that the most useful interpretation is possible. It would seem that the following criteria for representing and analyzing interactions must be satisfied.

First, each documented interaction between building blocks should be able to be understood in a way that is not dependent on subjective factors, and ideally must be recognizable such that these can be linked to coordination activities.

Second, it must be possible to structurally deduce which specific characteristics match a particular type of interaction. Being able to explain what coordination is actually required during a design process can only be beneficial, also because more appropriate coordination modes for handling a specific interaction can be suggested.

Third, it must be clear for each interaction how (by which technical decision) an interaction can be changed. This is needed to suggest options for manipulating an interaction such that future coordination is facilitated.

To sum up, to achieve a systematic and meaningful analysis it should be possible to define the interactions such that for each type of interaction it is possible to identify its cause, its impact on coordination, and the options for manipulation.

Surprisingly little research is available on this topic. The investigations (both general and specific) described in this chapter are valuable but lack sufficient detail to be of direct use for this research. The detailed study by Sosa and Eppinger seems to be the most promising, but it was concluded that their way of representing product architecture was not sufficient for an effective analysis. A lot of work remains to be done in both the practical and theoretical arenas. Accordingly, the following specific research questions have been formulated:

*Is it possible to identify various types of interactions between product building blocks such that it is possible to deduce the qualities of the system-like coordination required by each type of interaction, to understand the technical reasons for preferring one type to another, and to see what the options for manipulations are?*

The proposed representation of product architecture will be applied to a real-life case. Based on the results, how the types correspond to system-level coordination activities will be explored, as well as the options for improvement they generate. The questions below refer to the practical validity and usefulness of the proposed representation:

- Is the representation of the architecture generally understood, and can each interaction be linked to system-level activities?
- What system-level coordination activities match each type of interaction, and are the premises behind the coordination characteristics per interaction type valid?

It is to be hoped that these findings will ultimately lead to a Thompson-like theory that is able to identify the consequences for system-level coordination based on a representation of interactions within the product alone.

- Does the analysis result in options to improve the design process, and what are these options?
- What is the effect of the implementation of these options on project performance, or at least, how can these be measured in future?

The first question will be addressed in Chapter 5, the case study will then be described and discussed in Chapter 6 and the study will be summed up in Chapter 7.

# 5 Proposed taxonomy

This chapter represents the main contribution of this research to the available body of research linking architecture and organization. A new taxonomy of interactions between product building blocks will be proposed, one that can be viewed as linking engineering and organizational knowledge. The taxonomy is designed such that the characteristics of system-level coordination can be understood, and basic insights into options for improving the design process are provided. Broadly speaking, it is a means for understanding system-level requirements during the design processes by analyzing the structure of an existing product. The representation of a particular product architecture must trigger the generation of more suitable coordination devices that fit particular types of identified interactions, or suggest how the most difficult interaction can be manipulated in order to reduce future need for coordination.

The interaction types that will be introduced are built upon well-known architectural definitions and constructs. In that sense the taxonomy is not new. However, compared to available taxonomies the categorization and choice of constructs makes it much more effective for analyzing and combining available knowledge. The taxonomy and its features will be introduced according to the following framework:

- Introduction to the taxonomy's main concept.
- Proposal and discussion of the three types of interaction.
- Linking of each type of interaction to coordination.
- Discussion of the taxonomy.
- Summing up.

## 5.1 Introduction to the taxonomy's main concept

To this point, the idea of documenting interactions between building blocks as a useful way of representing product architecture has been proposed. Identifying the interactions between all possible pairs of building blocks will provide a clear overview of product architecture. Moreover it will provide an opportunity for analyzing system-level coordination, if it can be assumed that the structure of the project matches the structure of the product. Interactions between building blocks can be considered as the rationale behind system-level coordination, and understanding such a rationale will in turn promote understanding of what system-level coordination is actually required.

In the previous chapter it was argued that the effectiveness of such an analysis is critically dependent on what types of interactions can be identified between building blocks. For a useful analysis it must be possible to recognize what the technical reason or reasons behind an interaction are, and by which technical decision(s) an interaction can be manipulated. Moreover it is essential to be able to deduce the impact of a type of interaction on the system-level coordination characteristics. The question that remains is what these types look like. For a solution, this chapter will reexamine and link some of the important findings in Chapters 2 and 3 of this thesis.

First, Ulrich's original definition will be looked at anew. According to him, three types of technical decisions determine product architecture. These can be put under the headings of functional arrangement, mapping from functions to building blocks, and specification of the physical interfaces. It will be argued that since these decisions determine architecture, and architecture is about interactions between building blocks, each of these decisions (and these decisions only) determines the interactions between building blocks.

This chapter will demonstrate that the role of architectural decisions is to achieve good integration between the building blocks. In fact, each decision can be interpreted as a set of conditions, technical and otherwise, that need to be satisfied by the building blocks in order to create a properly functioning final product. Each architectural decision refers to different technical conditions and can be thought of as a type of interaction. The proposed taxonomy must thus facilitate translation of the technical architectural decisions into types of interactions between building blocks.

This has at least two advantages:

- We will know the underlying reason (which technical decision) for each type of interaction.
- We will know how an interaction (by which technical decision) can be manipulated.

In addition, as will be shown in 5.2 the 'architectural' decisions can be used to identify the coordination characteristics. For each type of interaction it will be decided whether goal-setting (as specified in Chapter 3) between the teams is possible or not. The main idea here is that the hierarchical structure of the technical decisions (design goals versus physical solutions) can be linked to the hierarchical concept of goal-setting (goal or task versus detailed activities/decisions) between teams.

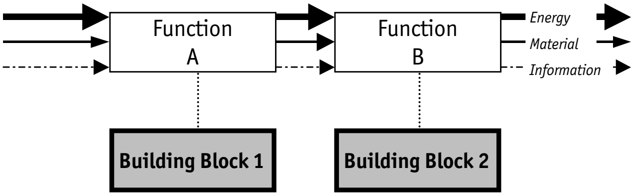
In the next section a taxonomy of three types of interactions between building blocks will be introduced: the functional, the mapping, and the physical type of interaction. These definitions will be based on the technical constructs as proposed in Chapter 2.3. Interpretation of the interaction types will be inspired by the prescriptive design methodologies, in particular axiomatic design.

## 5.2 The three types of interactions proposed

Under the following headings, the three types of interaction will be described in the order previously introduced. Each type will be discussed separately. In the discussion the relationship between the types will be addressed.

### *5.2.1 The functional type of interaction between building blocks*

Two building blocks have a functional type of interaction when their functions are connected by flows of energy, material, or information. Figure 5.1 shows this particular interaction type. A change to one building block that affects the specifications of its functional output is sufficient to force a modification of the specifications of the required functional input of another building block, and vice versa. This interaction type clearly relates to functions that can be expressed as inputs and outputs of energy, material, or information, and is neutral in respect to particular physical building block solutions. This interaction type is identical to the relationships described in the functional scheme of Pahl and Beitz or Ulrich and Eppinger. However, it differs from the exchange type of interactions of Pimmler and Eppinger since physical realization is not included.



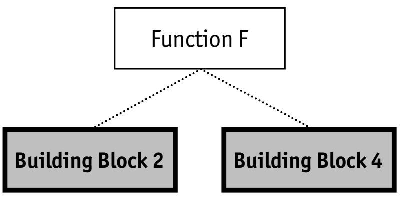

▸ **Figure 5.1** The functional type of interaction

### *5.2.2 The mapping type of interaction between building blocks*

Two building blocks have a mapping type of interaction if they are mapped to the same function that is not initially decomposable, in line with the discussion in Chapter 2. This type is illustrated in Figure 5.2 (that is based on the example of Figure 2.12). Accordingly, a change to one building block that affects realization of the shared function requires a change to the other building block to properly realize their function, and vice versa. The building block physical characteristics will together result in a working interrelationship (Pahl & Beitz 1996) that fulfills the function.

A building block has no mapping interaction if its mapping (from functions to building blocks) is one-to-one or N-to-one. In these cases, the block fulfills one or more functions itself without direct interference from alternative blocks.

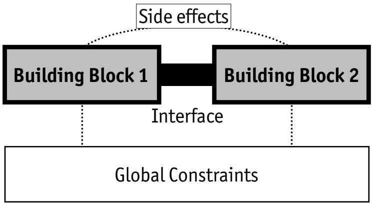
This type of interaction is obviously derived from mapping from functions to building blocks (according to Ulrich), and is also similar to the theory of axiomatic design. Nevertheless, available taxonomies do not distinguish the mapping type of interaction from the functional one. This distinction will be dealt with in the discussion.




▸ **Figure 5.2** The mapping type of interaction

### *5.2.3 Physical interactions*

In addition, there are physical interactions between the building blocks since the blocks have to be physically put together (assembled). In contrast to the previous two types, these do not directly refer to desired functionality. Three sub-types can be identified: interactions due to a physical interface, global constraints, or side effects. These are all based on 'interface coupling' that according to Ulrich greatly affects modularity. Figure 5.3 shows the three sub-types, which are described below.




▸ **Figure 5.3** Physical interactions

*Interfaces*

Two building blocks physically interact if a change to one that affects the other is necessary in order to realize their interface. This interaction not only refers to a physical connection but also (where such is relevant) to the physical realization of a functional interaction.

*Global constraints*

Physical building blocks physically interact if they are subject to the same global constraint. Accordingly, a useful change to one building block that results in exceeding the global constraint requires a change in the other block in order to satisfy the constraint. For instance, each building block has a size, shape, and position, and the use of space excludes the use of another building block. If the total space is determined (limited) then the blocks interact. Because all building blocks interact to some extent in this way (they all contribute to weight, space, and so on) only those useful changes to building blocks that have a significant (to be specified) impact will be considered here.

Available taxonomies do not include global constraint interactions. Spatial interactions (Pimmler & Eppinger 1994), or a need for orientation (Sanchez 1999a) can, however, be a consequence of this interaction type.

*Side effects*

Two building blocks physically interact if a change to one block that affects its side effects requires a change to the other in order to function correctly. Building blocks generally generate heat, vibration, magnetism and so on as a side effect of the design parameters for realizing a desired function. Since these interactions depend on detailed physical parameters or combinations of some (i.e. position) in respect of both blocks, and are difficult to define in a function structure, side effects can best be categorized as a 'physical' type of interaction.

It should be noted that the three sub-types have fundamentally distinct reasons for interface coupling. These reasons do not come into Ulrich's definition but are included within the corresponding examples discussed in Chapter 2.

### *5.2.4 The relationship between the types: discussion*

To this point the three interaction types have been introduced separately and they will now be discussed in relation to each other.

All interaction types should be considered as equally important. The fact that the global constraint interactions were dealt with relatively late does not necessarily mean that these should always be considered at a later stage than the others. Not all interaction types, however, are fully (sequentially) independent in the sense that one type of interaction also causes another type of interaction. This will be described below.

The functional and the mapping types are independent. Both result directly from the mapping of the functional scheme to the building blocks, but a functional type does not imply a mapping type of interaction between two blocks, or vice versa.

The physical interface interaction depends on the functional and mapping interactions. If there is a functional or a mapping interaction between two blocks there is also by definition an interface interaction that establishes the physical realization of the exchange. Conversely, an interface interaction between two blocks does not necessarily imply a functional or mapping type of interaction between two blocks. For instance, two blocks that are physically attached do not necessarily have a 'functional' relationship (think back to the example of the bottle and the cap in Chapter 2).

Due to global constraints, the physical interaction does not have a relationship to the other types. Note however, that realization of a mapped or other function may indirectly affect global constraints. For instance, the aesthetic design of the shaver housing may cause a limited amount of space for remaining building blocks.

Finally, due to side effects the physical interaction does not have a relationship with the other types. In the final discussion of the taxonomy, the dependencies between the types will be looked at in greater detail.

## 5.3 Impact on coordination per interaction type

This section will illustrate how the interactions of an existing product can in theory be translated into system-level coordination characteristics. The following illustration will throw light on the rationale behind this.

Essentially, a project team has the task of defining a whole range of detailed design parameters that together satisfy all product functions and constraints. When the project team is split up into smaller design teams, each of these teams has the task of specifying the detailed design parameters necessary to establish a physical building block. Obviously, if all of these design teams do their work in complete isolation it is extremely unlikely that their joint achievements will result in a functioning end product. System-level coordination is needed to compensate for the fact that the teams will act as if they are fully independent.

Furthermore, it is also evident (after Galbraith) that it is impossible to draw up a detailed protocol that specifies what detailed decisions each team has to make in each situation in order to realize the overall product. Goal-setting is a beneficial mechanism for achieving coordination. Teams ideally specify goals at a relatively high level of abstraction and are able to implement their decisions concurrently. As was argued in the chapter on organization (based partly on the work of Thompson), whether it is possible to apply goal-setting will depend on the conditions that are included by an interaction. The more conditions there are and the more detailed they are, the fewer conditions for constructing a building block will be independent of what is done by other teams.

Because architecture is being mapped on organization, these conditions can be identified with precision. In fact, the 'conditions' that need to be coordinated are embedded in the interactions between the building blocks of the product. The more detailed the interactions between the building blocks are, the less the ability to apply goal-setting, and the more intense system-level coordination becomes. This section will consider to what extent goal-setting is possible for each type of interaction. For this, chapter 2 will be largely relied on, and the prescriptive literature directed at efficient design.

It should be noted that a pair of building blocks may engage in multiple interactions (of several types). In this section, however, the characteristics of each type will be described as if these are the only ones between those blocks. In the event of multiple interactions occurring, each of these interactions has to be coordinated and the consequences are (at the very least) additive. In the discussion the possibility of the various interactions interfering with each other will be considered.

Finally, it should be stressed that the aim is not to explain the system-level coordination effort from the very beginning of a design project. It will be assumed that for each product that will be analyzed its functions, mapping, building blocks, and constraints are known. In fact, the period analyzed will start from after the architecture has been determined and extend to the final design (see the discussion at the end of this chapter).

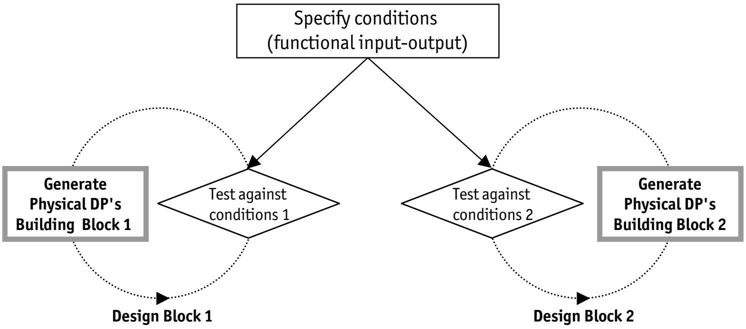
The impact on coordination will be described below for each interaction type in the same order as introduced in the previous section. For each type there will be a brief reference to how the goals between the teams are set (and communicated), to what extent the teams can work in parallel, and a brief note on the likelihood of exceptions to the planned goals (similar to Galbraith's theory).

### *5.3.1 Functional interaction and coordination*

Two teams whose blocks are engaged in functional interaction are able to apply goal-setting to a considerable extent and require little system-level coordination. Both teams can make detailed design decisions in full isolation from each other as long as they each achieve the appropriate functional specifications (inputs and outputs). A power supply design team, for instance, is allowed to do what it likes as long as the design establishes the appropriate specifications for energy output.

A functional scheme of specification and agreement can, in fact, be described as specification of the goals that each team has to meet. It should be noted that the generation of such a scheme (specification of the goal itself) may involve many iterations and require intense communication (Pahl & Beitz 1996) to find suitable goals that can reasonably be expected to be achieved by the teams.

Figure 5.4 conceptualizes the coordination of a functional interaction. Once a functional scheme has been devised it can be readily communicated to the teams involved. The teams can each try to generate and test detailed design parameters concurrently in order to achieve their 'goal'. As long as the design of each building block meets its planned specification no additional coordination is required and the teams can work concurrently. One team failing to reach its planned goal may have a disruptive effect. In that case additional system-level coordination is required to solve this problem. In general, the more 'difficult' it is to meet a functional specification, the more failures may be expected to occur.



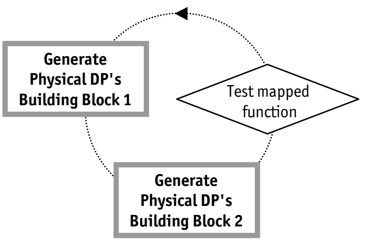

▸ **Figure 5.4** Coordination of a functional interaction

### *5.3.2 Mapping interaction and coordination*

Two teams whose blocks have a mapping type of interaction will be hampered in their goal-setting to a certain extent. Since their blocks have to jointly achieve a function, detailed design decisions cannot be made in full isolation of each other and mutual adjustment is a prerequisite.

Coordination of a mapping interaction relates to agreement on a functional scheme where two blocks need to jointly satisfy one function. Each team's particular goal specification will include very detailed specifications of design parameters that each team needs to realize. In effect, the search for these separate goals will go hand-in-hand with the jointly evolving design of the two blocks.

As shown in Figure 5.5, generate-test cycles will occur between the two design teams and they have to exchange information at a highly detailed level. Based on the logic of axiomatic design, the following consequences can be deduced. In order to find an appropriate setting for both blocks, team 1 specifies a set of design parameters that (in their judgement) provide a contribution to the joint function. These specifications must be communicated to team 2. Doing so may be very complex since the detailed design parameters chosen may include multiple and complex expressions of detailed characteristics, possibly with hard-to-communicate sensitivities and behavior (Whitney 1996). Subsequently, team 2 adds to the design parameters of block 2 in such a fashion that it is expected that the blocks will jointly fulfill the function. However, whether the function is correctly fulfilled cannot be seen on paper but has to be tested in collaboration with the other team. If the results are negative the cycle starts again. If the results are satisfactory the required goals for each team are specified separately, but at the same time these are also realized in the relevant design parameters of both blocks.




▸ **Figure 5.5** Coordination of a mapping type of interaction

When for some reason (for example, other goals having to be met) one team fails to meet its design goal, then additional system-level coordination is required. Since this type of interaction includes many detailed conditions, failure is likely to occur, as was concluded with respect to Thompson's work.

It should be noted that the way that the cycles are conceptualized here is somewhat primitive. The cycles will probably be performed within multiple layers, i.e. first the selection of a working principle and then the selection of the detailed design parameters, as in the work of Pahl and Beitz. In any case, this reasoning highlights the expected need for considerable mutual adjustment between teams if they have to deal with a mapping type of interaction. It would appear that such an interaction differs in this respect from a functional interaction where specification of the goals and realization of the goals can be cleanly separated for each block.

### *5.3.3 Physical interactions and coordination*

The impact on coordination of the physical interaction type will differ per sub-type. Except where there are global constraint interactions, it is more difficult to generalize about the coordination effort required. Globally constrained interaction will thus be dealt with first and then the side effects and physical interface described.

#### *Global constraints and coordination*

Two teams whose blocks are involved in a global constraint interaction with global constraints are able to apply goal-setting. A global constraint can be decomposed into a constraint for each block. As long as each team satisfies this constraint, the teams are able to make all detailed design decisions independently of each other[3].

Specification of all sub-constraints can be very difficult and require considerable coordination effort, though once specified, the conditions can be easily communicated across the organizational units, and it is easy to check whether each blocks is staying within the constraint. As long as the design of each building block fits within its planned sum of space, weight or costs, the design can be executed concurrently and the overall constraints are satisfied. The tighter the constraints for each block the more difficult it obviously is to satisfy

[3] Note the similarity between functions and global constraints here. In fact, both can be decomposed into 'smaller elements' that can each be considered as a 'design goal' by each of the teams. As a logical consequence of this assumption, it is possible to think of a strategy where two teams have to jointly satisfy (stay within) the same constraint. In such a case, decomposition of the constraint has to take place at a low level of detail in close team collaboration, similarly to a mapping interaction. Nevertheless, mapped constraint interactions will not be considered here. The reason is that technically speaking, global constraints are easy to decompose, in contrast to (mapped) functions. Global constraints thus allow for goal-setting. This is not to say that it is easy to find a decomposition of a global constraint that is appropriate (realistic) for each building block.

the conditions, and the more likely the occurrence of failure and additional system-level coordination effort.

*Physical interfaces and coordination*

Two teams whose blocks have a physical interface need to perform system-level coordination at a detailed level. However, it is difficult to generalize about such an interaction. Because it has so many physical aspects, this type of interaction probably has many of the characteristics of a mapping type of interaction. The problem, however, is that it may be difficult to locate a physical interface. Can it be seen as a part of one of the two blocks or is it to be considered a shared feature? Testing of an interface will obviously require two blocks and has to satisfy a large number of production and assembly constraints and wishes.

*Side effects and coordination*

The coordination of side effects involves reacting to unplanned or unintended effects. According to design literature (Pahl & Beitz 1996, Ulrich & Eppinger 2000), management of side effects involves close coordination between the organizational units involved and requires trial-and-error testing of small changes to physical parameters. A side effect may be considered an unexpected consequence of the original specifications and, as a logical consequence, cannot be planned for. It is simply impossible to set any rules for such occurrences. The amount or type of coordination will depend on what type of specification has not been met.

## 5.4 Discussion of the taxonomy: its role, comparison, and restrictions

A taxonomy of three types of interactions between physical building blocks of a product has been proposed. The role of the types will be briefly considered below, and the alternative taxonomy (of Pimmler and Eppinger) looked at a second time, and its restrictions highlighted.

### *5.4.1 The role of the taxonomy*

The functional, mapping, and physical types have been delineated on the basis of Ulrich's general definition of architecture. The underlying reasons for each interaction are embedded in the definition of the constructs, and the available literature and knowledge of product architecture can be used to interpret each interaction type. This can be expected to be beneficial for (1) a general understanding of what interactions involve (2) generation of options for improvement, and (3) indicating the contingencies inherent in any structure where interaction takes place.

It will first be argued that understanding the technical reasons behind an interaction will increase team member understanding in general. As such, the way that they perceive the interaction will not depend on background factors, temporary matters or individual factors.

Second, insight into the background to the interactions provides useful information on how an interaction can be manipulated and what the potential difficulties are. In fact, changing the structure of interactions means that the technical decisions that give rise to the interactions need to be altered. For instance, a mapping interaction can be manipulated by changing the way functions are mapped to building blocks.

Third, understanding the contingencies inherent in an interaction structure can be obtained by structurally discussing the why of underlying decisions. As argued in 2.4, there may be many considerations underlying an architectural decision that subsequently gives rise to an interaction between building blocks. Some of the interactions may thus be seriously embedded in broader contingencies such as production structure, available technology, unit cost price, special priorities, or traditional ways of problem solving. Consequently, it is not only imperative to show how an interaction can be technically manipulated, but also how manipulation may be hampered by much broader considerations.

How each type can be individually manipulated will not be dealt with any further since this will depend on the definition. In the case study, though, this aspect will be thoroughly illustrated.

It has been shown that it is theoretically possible to deduce the characteristics required of the system-level coordination that matches each type of interaction. Prescriptive design models aimed at efficient design provide the framework to do so. Real-life projects, however, are generally not as clear and pre-structured as the prescriptive models. Hence engineers may apply more intense or different ways to handle interactions, but these are never less than the prescriptive characteristics. These minimum requirements can be expressed in terms of propositions. Their role is twofold. In the first place, they represent a step in the direction of a Thompson-like theory able to explain the differences in system-level coordination effort during a design process, based on the characteristics of the product alone. Second, they provide guidelines for understanding how the management of a specific interaction can be improved. On the one hand, actual coordination devices may be altered such that these correspond better to the propositions (ideal or less so) per type of interaction. On the other hand the propositions can be used to illustrate which effort is by definition embedded in an interaction, and hence to indicate when further improvement of coordination cannot be reasonably expected without changing the interaction.

To sum up, it has been argued that the proposed three types of interactions satisfy the criteria that were suggested in the formulation of the problem in the previous chapter. The taxonomy will now be compared to that of Pimmler and Eppinger (1994) so as to demonstrate this taxonomy's particular advantages.

### *5.4.2 Comparison with Pimmler and Eppinger*

When the types of interaction were defined, it was mentioned that there were some differences with those in the taxonomies in the literature. Using the taxonomy of Pimmler and Eppinger, these will be illustrated in greater depth and the consequences discussed.

In Chapter 2 it was mentioned that the taxonomy of Pimmler and Eppinger is not easily translatable into Ulrich's architectural decisions. In contrast to the interaction types in this thesis, their taxonomy does not provide a clear decision-making hierarchy. The exchange types may refer to functional aspects as well as physical aspects (the physical realization of the interface). Moreover, spatial interactions seem to refer to a physical solution (the location of a building block/component). Consequently it is not possible to deduce what decision resulted in a constraint on location. This may have involved a decision at a high level of abstraction (i.e. managing a global constraint) or at a more detailed level (i.e. a solution to a mapping interaction).

Each of the constructs thus may refer to different levels of detail and it is not possible to make a statement about the required type of coordination. It should also be remembered that poor alignment with the architectural decisions hampers clear understanding and the generation of options for manipulation.

Most remarkably, though, is that in contrast to the taxonomy proposed here, Pimmler and Eppinger do not include a mapping type of interaction. It has been shown that introduction of the mapping type of interaction contributes to a meaningful analysis of coordination requirements and options for manipulation. The question thus remains as to how existing taxonomies model mapping interactions. There are two possible explanations.

The first may be that the spatial interaction takes care of all mapping interactions. However, this does not seem adequate since many more detailed parameters than proximity alone may be involved in the fulfillment of a function. Moreover, proximity may even be a result of the exchange types (the physical realization of exchange of material energy or information). When, for instance, an interface between two blocks is fixed, then these must also be in proximity to each other.

A second explanation may be that mapping interactions are modeled by exchange interactions. Within the taxonomy in this thesis, the distinction between the functional and the mapping types is simply based on the assumption that a function cannot be fully decomposed (to the most detailed level) without considering 'physical solutions' (as concluded in Chapter 2). If for argument's sake it is assumed that (after Pahl and Beitz) functions can be decomposed in a solution-neutral fashion, then a mapping type can be written as a collection of functional interactions at a lower level of abstraction. A mapped function is split up into such levels of detail that each sub(sub)function (transformations of input and output) can be completely allocated to a block. As a consequence, the mapping type of interaction is transformed into a set of input-output exchanges between the sub(sub)functions of both blocks. Perhaps this reasoning explains why available taxonomies mainly consist of exchange types and do not include mapping types of interaction.

However, in line with the conclusions in Chapter 2.2, decomposition of functions depends strongly on chosen physical solutions. As a result, expressions of input and output between sub(sub)functions strongly depend on detailed physical characteristics and probably involve many and complex expressions of input and output (including force). Furthermore, many design goals are practically impossible to decompose. As a result it will be argued that the introduction of the mapping type gives this taxonomy a clear advantage over other available taxonomies.

To sum up, the proposed taxonomy provides a contribution to available taxonomies because, unlike those in the literature, the types of interactions described here can be linked to architectural decisions.

### *5.4.3 The taxonomy's limitations*

Three comments on the proposed taxonomy should be made. The first involves the fact that building blocks are considered as being black boxes, the second deals with the fact that only the final interaction structure is dealt with, and the third has to do with the lack of time and cost issues relating to these constructs.

#### *Black box*

This taxonomy of interactions is purely for indicating interactions *between* building blocks. This does not mean that no interactions exist *within* the building blocks. The contrary is true. The interactions within the blocks are generally much stronger than the interactions between the building blocks. The inner interactions are not modeled, however, since we are only interested in system-level interactions and coordination. The blocks are thus considered as being black boxes. The limitation that this imposes is that it is not possible to define potential relationships between the types of interactions at a lower level of abstraction. While it was previously argued that a side effect interaction does not have a relationship with the other types, at another level specific design parameters that are required to fulfill a functional interaction may also refer to a side effect. A single design parameter may thus play a role in multiple interactions. The way in which one interaction is satisfied by a design parameter may thus have an impact on the way another interaction can be solved.

Since including all these details would have been at the expense of a general overview of system-level interactions, these aspects will not be taken into account within the current taxonomy. During the case-study discussion, this issue will be looked at in greater depth.

*Final interaction structure*

Product design is an evolving process, and consequently interaction structures may be expected to change during the course of the design project. As a result, there may have been system-level coordination of interactions that are not included in the final set of interactions preceding completion of the documented product.

The design of a product is an evolving process, and as a result the structure of interactions probably is dynamic during the course of the design project. Consequently, there may have been system-level coordination effort for interactions that did not return in the final set of interactions of the almost finished product that we documented. Since coordination is mirrored on the final set of interactions a portion of coordination that took place maybe missed. However, it is argued that (after the architecture was specified) most of the types of interactions are the same for the whole period and only the exact specifications were subject to dynamics. The final set will therefore be reasonably representative of the process as a whole.

Furthermore, it should again be stressed that the coordination activities that took place to define the basic architectural decisions have not been addressed. According to Henderson and Clark (1990) this involves a period of considerable experimentation and intense coordination and it is in fact impossible to speak about building blocks at all. For this analysis, what this implies is that for projects involving radical innovation (see Chapter 3), only a small part of the coordination of the whole design project can be analyzed. However, for projects involving incremental design, a large part of the processes can be analyzed since the major architectural decisions will already have been made early in the project.

*The interaction constructs only define the technical conditions*

This taxonomy of interactions applies to the technically inevitable system-level coordination that takes place between the design teams. In addition, there are many other aspects that affect system-level coordination. Besides resulting in a technically correctly functioning product, the project has to be finished within a specific schedule and has to be feasible in terms of the financial budget and number of designers. To that end, design teams also need to satisfy goals with respect to these issues. The tighter the restrictions, the higher the probability of failure, and the more system-level coordination probably is required. However, this taxonomy does not include such managerial factors. The focus of this study is solely on the unavoidable underlying technical product structure. This issue will be briefly reexamined in the case-study discussion.

## 5.5 Summing up and field of application

To recapitulate, in contrast to the available taxonomies, a taxonomy of technical interactions between building blocks that can be clearly linked to architectural decisions has been formulated. The benefits of the proposed interaction types are that they facilitate understanding of the reasons for documented interactions, and options for manipulation can be easily derived. Moreover, the impact on the coordination of each interaction type can be derived from the characteristics of the underlying technical constructs.

Perhaps it is no accident that the variety of technical decisions contained in the prescriptive design literature can be translated into coordination characteristics. The 'layers' of technical decisions not only have to facilitate effective problem solving, but must also facilitate structured coordination across designers. In fact, if the various technical constructs were not suitable for the working together of a great many people, the prescriptive literature cannot have been based on good practice.

Finally, those situations in which the suggested approach is likely to be the most appropriate one will be described. Although it can be argued that the technical theories will be valid across a whole range of physical products, some conditions can be specified for which the interaction approach is particularly relevant. These include:

- Products that are large and complex enough to be decomposed into building blocks.
- Products that consist of all three types of interactions (not fully modular products).
- Project teams where many designers are involved, and the design teams are best organized around the product building blocks.
- Design projects that are under pressure to improve their performance.

It would appear that in those cases where these four aspects apply, the theoretical constructs are particularly relevant. This is not to say that the theories are not appropriate for fully modular products. However, a focus on interaction is far less useful since very few interactions (and none of the mapping type) will occur between the building blocks and these are thus a variable of little relevance in improving design processes.

The question now remains as to how the proposed constructs will apply in practice. This will be illustrated and tested in the next chapters.

# 6 The results of the case study

The foregoing chapters of this thesis have stressed the importance of understanding and improving system-level coordination during a design project. In addition, it was argued that the reasons for system-level coordination during a design process are embedded in a product's architecture. It was therefore suggested that system-level coordination during a design process could be analyzed and improved based on a detailed representation of the particular architecture of a product. In order to increase an understanding of this relationship and to facilitate its exploration in practice, a taxonomy of interactions between product building blocks was developed (Chapter 5). The taxonomy identifies three types of interactions, each linked to system-level coordination characteristics. The most important input for this taxonomy was provided by a combination of prescriptive engineering design theories described and discussed in Chapter 2.

This chapter will now illustrate application of the taxonomy to a single case study. The system-level coordination of an electric shaver design process was analyzed from the point of view of its underlying product architecture and a significant number of potential improvements identified.

The case study was conducted for a number of reasons. The most obvious one is that an illustration of the types of interaction involved in the design of a real product places all the technical and theoretical considerations of the previous chapters into a real-life context thus enhancing understanding of the concepts within that particular context. The other reasons are more directly linked to the research goals formulated in Chapter 4:

- To check the validity (objectivity) of the types of interactions.

In Chapter 3, the DSM models were criticized because their interaction constructs are heavily dependent on the personal focus and perception of the interviewees, which hampers objective and valid analysis. One of the goals of the previous chapter, therefore, was to create a taxonomy of interactions for which the underlying reasons are clear. A real-life situation to find out whether this produces valid and generalizable results is obviously needed. The results of this will play a leading role in the interpretation of the remaining case study findings.

- To illustrate how each type of interaction reflects system-level coordination during the design process, and to explore whether this matches the theoretical expectations of Chapter 5.

These results may be a first step towards a Thompson-like theory that, based on identification of interactions within a product, can pose propositions about the coordination effort required during a design process.

- To explore whether and how the analysis contributed to understanding and to the generation of options for improving the design process.

In Chapter 5 it was suggested that the proposed taxonomy would be useful for improving the design process in such a way that (1) the coordination mechanisms could be adjusted to better match the characteristics of a specific interaction, and (2) that the interactions between the blocks could be manipulated in order to reduce the need for system-level coordination. Accordingly, the role of this taxonomy in guiding a project team towards understanding and

improvement needs to be explored in practice.
- To explore the claim that these options actually improve the performance of the design process.

This chapter is arranged into three main sections: research setting, results, and discussion of the results. The first section describes the research setting required to achieve the research objectives. It deals with the methodological aspects of case-study research, including a detailed description of the shaver case and the data-collection protocol.

The second section describes the case study findings. The interactions that took place during the design of the electric shaver will be described according to the proposed taxonomy. This will be followed by a description of how each interaction mirrors system-level coordination activities during the shaver design process. There will then be a detailed analysis of how the interactions and related coordination demonstrate the company's current way of working, and how this has resulted in a significant number of options for improving shaver design processes. Finally, there will be a brief exploration of how these options for improvement actually affect the company and to what extent this can be traced in improved project performance.

The third and last part of this chapter discusses the results of the case study and examines whether the research goals have been met. These issues will be dealt with under the following headings:
- Research setting.
- The interactions between the building blocks.
- System-level coordination .
- Options for improvement.
- The effect on performance.
- Discussion.

## 6.1 The research setting

This research opted for a single case-study design as discussed in Chapter 1. According to Yin, a single case study is an appropriate way of confirming, challenging, or extending theory on condition that the theory has specified a clear set of propositions as well as the circumstances in which the propositions are believed to be true.

The present section addresses the case study setup. It is argued that the role of theory, the case circumstances, and the data-collection protocol jointly contribute to a reasonably valid and reliable achievement of the research goals. The starting point is a description of how the case study is embedded in the theoretical constructs of the previous chapters. The shaver case circumstances will then be discussed in order to place the results of the study in context. Two data-collection tactics that increase the validity of the investigation are then defined and the data collection protocol specified. Finally, the research setting will be reexamined in order to ascertain whether the requirement of good scientific quality has been met.

### *6.1.1 Case study setting: theory*

Despite the case study mainly having an exploratory character, the role of theory cannot be underestimated. The theoretical part of this research constitutes an important part of the thesis in its own right and as an underpinning of the practical aspects. According to Yin, 'the use of theory, in doing case studies, not only is an immense aid in defining the appropriate research design and data collection, but also becomes the main vehicle for generalizing the results of the case study.' This section briefly explores how the theoretical constructs were used when collecting the data, and how the findings may be generalized. Furthermore, the circumstances in which the theories can best be applied will be listed.

The case study will explore how a detailed representation of a particular product architecture leads to understanding and improvement of system-level coordination during a design process. The taxonomy of interactions was developed prior to the case study and played a major role in data collection. First, the documentation of the shaver interactions was completely determined by the three types of interaction. Second, system-level coordination was analyzed in terms of the interactions of the technical reasoning behind the linking of interactions to coordination characteristics. However, collection of system-level coordination data was more open due to the explorative nature of the investigation. System-level coordination issues observed in practice were reported but not directly included in the prescriptive reasoning.

The taxonomy of interactions guided and focused the collection of the data to a large extent and provided the means for ensuring the validity of the results. The data collection will be further discussed in section 6.3.

The strong theoretical underpinning of the interaction constructs provides the basis for generalization of the findings across the shaver case context. Since the interactions are based on engineering design theories that have proven their value for a very broad range of products, analytical generalization can be applied (Yin 1994) to translate the results of this single case to other situations where the theories also apply. Note, however, that generalization of the results cannot be accomplished by analytical generalization alone and requires additional proof based on a multiple-case strategy. The findings must be tested in alternative cases where similar results should occur. This research could then be used as a starting point for further (multiple case) research.

After having addressed the importance of theory in achieving validity, there will now be a description of the specific conditions of the shaver case. The data-collection protocol, where the above remarks about the role of theory are revisited in greater detail, will then be illustrated.

### *6.1.2 Case study setting: the characteristics of the case*

This case study is conducted within a company that is a major player within the male shaver industry. The company designs and produces great numbers and a wide variety of electric shavers that are sold all over the world. The design processes of its most innovative product, a waterproof shaver with fashionable styling and an innovative shaving system, which has (as we know now) been a great success in the market place, were investigated.

The shaver industry and the design project under study will be described in this section. It will be argued that the specific characteristics of the design project (complex non-modular product, large design team split up into design teams, and high pressure on performance) are of particular relevance to this analysis of interactions. In addition, this section's detailed sketch of the case circumstances has the function of enabling generalization of the results for use in other cases.

To this end, the following will be presented: a general description of the market, organization, and product, an overview of the overall situation, and a close-up view of the particular design project. It should be noted that these characteristics refer to the situation at the moment of our investigation.

*The market*

The shaving market is a worldwide one and can be divided into two segments, each containing a few major players. The first segment includes firms that sell non-electric razors meant for so-called 'wet-shaving'. The second one consists of suppliers of electric shavers that are based on the 'dry-shaving' concept. The company that is the subject of this study has traditionally operated in the latter segment. Very recently, however, the introduction of shavers that combine electric shaving with 'wet-shaving' has altered the standard categorization.

Compared to the often-described volatile high tech markets (e.g. telecommunication) where ultra short time-to-market is a requisite for survival, the shaving industry is relatively stable. Nevertheless, the business has become much more dynamic over the years. The market demands more variety, fashionable styling, and superior technological performance. This has resulted in a greater variety of shavers, and an increasing number of state-of-the-art innovations. These aspects are obviously additional to the traditional key success factors of the industry, such as low unit cost, high quality and service, and optimal logistical performance.

*The organization*

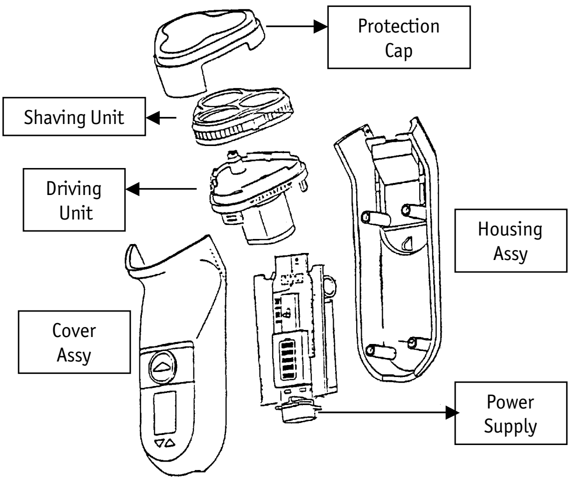
The plant that was this study's our focus is part of an overall corporate division that in turn is part of a large multi-national. The division headquarters house the shaving marketing department including product management and industrial design. The main departments of production, engineering, and product development are situated at, and under full responsibility of, the plant in this study.

Each of these departments is comprised of a number of units. In the product development department, the structure of the units is based on functional expertise: shaving technology, electrical technology, mechanical technology and so on. The organizational structure of the production and engineering departments is inspired by the structure of the product. Their units are organized around the main building blocks of the shaver. These so-called production units are highly mechanized and are built upon the mini-company concept. By means of these production units the company is able to produce a great variety of shavers based on the combination and assembly of the building blocks of the shaver. As a result, the plant is well-known for its high logistical performance.

New products are developed within design project units and managed by a project leader who is under the direct supervision of the plant manager.

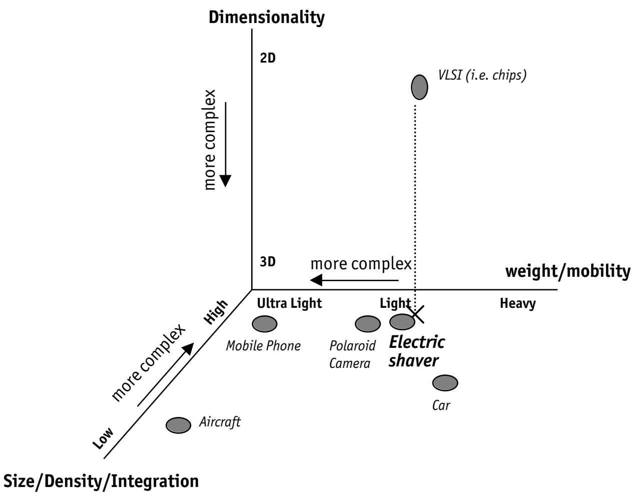
*The product*

The picture below shows a simplified picture of an electric shaver and its most common building blocks. A shaver consists of a protection cap, a shaving system, a driving unit, a power supply, and two covers.




▶ **Figure 6.1** The building blocks of an electric shaver

Electric shavers are high-volume customer goods. Technically speaking, they can be classified as electro-mechanic products. As the name suggests, this category of products combines electrical (e.g. the power supply) and mechanical (e.g. the shaving system) features, and these are well known for their complexity (Whitney 1996). In Figure 6.2, an electrical shaver is compared with other products based on Whitney's classification. This classification consists of three dimensions: size, dimensionality, and weight. Size and shape are extremely important for a shaver since customers do not like very large or cubic shavers. Furthermore, the various building blocks of a shaver interact in all three dimensions, and in that sense a shaver differs from VLSI design (that is mainly 2-dimensional). There are also limitations on the weight of a shaver, but this is not seen as a difficulty in the design process.




▶ **Figure 6.2** A shaver compared to other products

It should be noted that the products in Figure 6.2 are designed for worldwide use. They must function under all circumstances and satisfy all legislative requirements.

It should also be stressed again that shavers are high-volume products and thus unit cost cannot be dismissed out of hand. As mentioned in Chapter 2, this is a powerful factor that will play a part in the eventual design.

*The overall situation*

Under pressure from the more dynamic market circumstances, over the years the organization has significantly increased the variety and number of innovative products. As a result, the organization of product development has shifted from a single project organization towards a multiple project organization. The number of designers has increased significantly, design projects have become larger and technically more complex. These factors must all be juxtaposed against the limitations of low unit cost and the great importance of finishing design projects on time. Not surprisingly, the organization is facing increased organizational complexity. There is a great need for coordination and an increasing amount of time is spent on coordination activities.

*The design project*

The design project under study had to create a shaver with an improved shaving function and highly innovative styling. Furthermore, the shaver had to be washable, a new feature compared to previous ranges. The designers were at liberty to create 'optimal' solutions (subject to unit cost, shape, size, and performance) for this individual project as long as the building blocks of the shaver conformed with the production constraints. The members of the project team came from product management, several functional product development departments, engineering, production, quality, and so on. Because of the importance and complexity of this project a large number of people (X) were assigned to the core design team. The final number of people involved was even larger (4*X). This large team was then split up into so-called 'module groups' responsible for the design, engineering and production of a particular building block. In this thesis, design teams will be referred to instead of module groups (see also Chapter 2).

The project leader was responsible for the project in general, though strongly supported by the lead engineer who was responsible for the technical issues. The project team (including all design teams) was located in one room, which greatly facilitated mutual adjustment and informal communication. In addition, the relationship between the team and its environment (i.e. product management, general management, and the functional departments) was managed by means of formal meetings.

It can be concluded that the case clearly corresponds to the research purposes. The project matches the four conditions (specified in the previous chapter) for which interaction analysis is the most appropriate method. It has clearly identifiable building blocks, it is not entirely modular, the project team is large and split up into design teams, and there is considerable incentive to improve performance. The main characteristics are summarized in Table 6.1.

Attention will now be shifted to the data collection protocol and the rationale for conducting the case study will be established.

▸ **Table 6.1** Main characteristics of the company in the case study

| | |
|---|---|
| Business | Electric shaving, consumer goods |
| Market | Shifts from stable to more dynamic, but still focussed largely on unit cost |
| Production | High volume, great variety of end-products |
| Products | Complex electro-mechanic |
| Development organization | Multi-project |
| Project | Design new electric shaver |
| Project team | Large, cross-functional core team, split up into design-teams |
| Important features | Non-modular architecture, greatly improved shaving system, fashionable styling, washable |

### *6.1.3 Data collection: tactics and protocol*

It was mentioned in section 6.1 that the theories described in this thesis, in particular the taxonomy of interactions, provides the most important data collection input. Two additional tactics (as specified by Yin 1994) were employed to further strengthen the validity of the findings. After describing these tactics, the data collection protocol will be illustrated in depth. This will show how theory and tactics were both used to collect the data required by the investigation.

*Tactics*

In order to increase the validity of our findings, two tactics described by Yin were employed – multiple units of analysis, and multiple sources of evidence.

The first tactic is to consider the case as an embedded case study. Not only will the product as a whole be examined, each building block will also be individually studied. Of particular interest for this research are the interactions between each of the building blocks. Each building block is considered as a separate unit of analysis that can compared to one another. The interactions can be compared to each other in such a way that from a theoretical point of view, either similar results are obtained (literal replication), or results that are contrasting but for predictable reasons (theoretical replication). Many authors (Hartley 1994, Yin 1994) describe the benefits of these tactics for extensive analysis, enhanced insight and explorative power.

The second tactic is using multiple sources of information. Case-study findings or conclusions are much more convincing and accurate if the data collection protocol is based on multiple sources of information (Eisenhardt 1989, Yin 1994, Staudenmayer 1999). When these multiple sources of evidence all converge, the construct validity and reliability of the study increases. This tactic are called triangulation (Yin 1994). The sources of evidence used in this case are:

- Formal documents such as consolidation reports, design reviews, FMEA reports, QFD reports, and planning and meeting schemes.
- Interviews with many different designers from different backgrounds, project leaders, and management.
- The physical shaver itself.

Furthermore, to check the findings:

- Additional interviews with key informants were performed.
- The findings were presented to several groups within the company.
- A second evaluator assessed the material (Schoenmaker, 2000).

These features will be further addressed in the following section.

*The protocol*

The data collection protocol describes the procedures followed to collect data relating to the study's objectives. Each of the steps undertaken will be illustrated in the same sequential order as the investigation. Additional facts and experiences relating to each stage of the procedure are also included, in order to give a clear overview not only of which steps were taken, but also what these steps entailed. This aids assessment of the reliability of the findings (Eisenhardt 1989, Yin 1994).

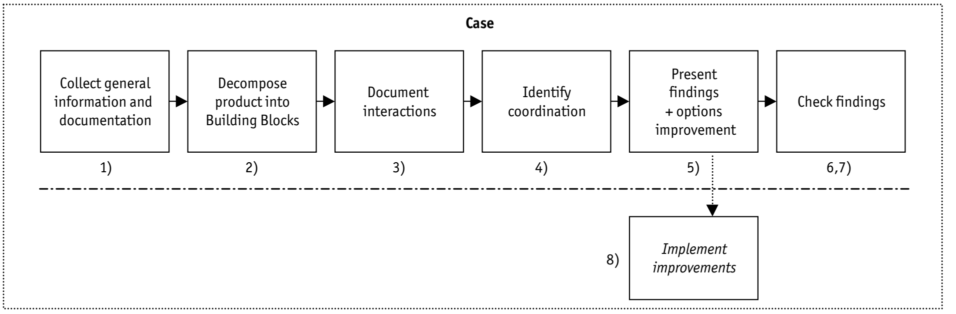
The data collection and analysis took place over a period of nine months, during which time the company was visited for four days a week. In the year that followed, the company was visited for two days a week, during which further investigations were carried out.

The investigation can be roughly divided into two phases. The first phase includes the period when the design process was analyzed and options for improvement were generated. The second phase commenced at the point of implementation of the improvements and the corresponding effects on performance.

Most of the data collection took place during the first phase. The analysis was retrospective. The product was an almost finished shaver, and system-level coordination was retrospectively analyzed. As a result the investigation could not have influenced the actual design processes. What it did was to introduce a new way of looking at the actual (and by that stage past) design process. During the second stage there was some involvement in the implementation of the options for improvement within new design processes. It should be noted that methodologically speaking, this is a completely different approach. Instead of analyzing a 'how is' situation, the new role was to collaboratively establish an improved design process with better project performance.

This thesis mainly focuses on the first phase of the investigation and uses the second phase as an illustration and exploration of the last research question, which addresses the effects on performance of the suggested options for improvement.

The steps performed are depicted in Figure 6.3 and described in the sections that follow. It should be noted that the dotted line in the Figure indicates the border between the analysis and the implementation of the results.




▶ **Figure 6.3** The data collection protocol

*1) The first step*

After having reached agreement on conducting the research within the shaver company, considerable effort went into getting to know the company, its people, the product, the design processes, and the underlying contingencies. A variety of documents were collected, including consolidation reports, design reviews, FMEA reports, QFD reports, and planning and meeting minutes. The investigation's protocol was then implemented.

*2) Decomposition of the shaver*

In close consultation with the project leader and the lead engineer, the shaver was split into ten physical building blocks. This decomposition logically followed from the division of work and the available production structure.

*3) Documenting the technical interactions*

The interactions involved in the design process were identified by documenting the interactions between each possible pair of building blocks (i.e. between A and B, A and C, A and D etc.). For each of the ten physical building blocks the most experienced designer available to be interviewed about the interaction between 'his' block and other blocks was chosen. There were thus two separate sets of observations (i.e. from A to B, and from B to A) for each pair of building blocks. Since interactions are symmetrical, these two observations could be compared. In addition, the project leader and the lead engineer were interviewed separately about all interactions. A total of 12 people with different backgrounds were interviewed and 4 different sets of observations for each pair of building blocks were obtained.

The interviews were semi-structured. Each interview started with the purpose of the research and the three interaction types being explained. The designer was then asked to describe all of the functions of the selected building block.

On the basis of this information the technical interactions between the selected building block and each of the nine others were identified. The designer described the interactions in his own words first, and he was then asked about each type of interaction, the questions being posed in a set order.

In order to minimize the chance of the designers having their attention distracted and to keep their focus on the interview as much as possible, the interviews were carried out in a separate room with a time limit of one hour. In most cases multiple one-hour interviews with the same person were required in order to achieve complete documentation of all the interactions.

After each interview the data was immediately processed and the document was returned the next day to the interviewed designer. When all the interviews were finished, the data was combined and returned to the designers in such a form that it could be compared and possibly adjusted.

*4) Identifying the coordination required for each interaction.*

During this step, the overall documentation of the interactions was discussed with each designer personally. The system-level coordination required for each interaction during the design process was then explored. These interviews had an open character, each designer being free to explain in his or her own words how each interaction was coordinated with the other design teams. Their answers were then structured according to the technical characteristics of each interaction and the propositions described in Chapter 6.4.

*5) Presentations and group discussion*

After documentation of the interactions and identification of the coordination required, the data was combined and an overall (retrospective) analysis was made of the system-level coordination during the design project from the perspective of the underlying product architecture of the shaver. In addition to structuring all of the data, a number of potential improvements for each interaction based on the prescriptive logic of the taxonomy were identified. The findings were reported in a highly detailed document, and a powerpoint presentation was constructed, including the following issues:

- General explanation of the relevance and purpose of the research.
- Explanation of the types of interactions and their importance.
- Illustration of the documented interactions between the building blocks of the shaver in a separate matrix per type of interaction.
- Illustration of the identified coordination effort required by each interaction.
- Suggestions for improvement per interaction. These were divided into opportunities for achieving more appropriate coordination mechanisms, and for technically adjusting the interactions in order to facilitate future system-level coordination.

These were presented to four different groups within the organization: project managers, two functional units within the product development department, and the general development management team.

The overall aim of the presentation was to show the company which possibilities for improvement could be identified based on an analysis of their current product architecture and way of working. The main purpose was to see whether they agreed with the analysis, how and in which way their understanding was increased, and to further extend the analysis based on their remarks and impressions. More precisely, during the presentation:

- The correctness and understanding of the documented interactions was checked.
- The correctness of the identified coordination effort was checked.
- Reactions to the analysis were noted, as well as whether the company agreed with the proposed improvements.
- The current way of working was discussed further, as well as other options for future improvement.

The presentations each lasted for more than two hours. During the presentation people were free to ask questions or give remarks about the analysis. At the end there was between half and one hour available for discussion and reflection.

*6) Interviews key informant*

After the presentations, the results and how they had been interpreted were checked and group discussions were held with two key informants. These key informants were highly experienced and had a good overview of the organization as a whole. There was a particular focus on how the interactions were generally viewed, whether the fundamental characteristics of the interaction types during the design process were recognized, and how coordination could be improved based on these understandings.

*7) Second investigation by a different evaluator*

In order to establish additional (unbiased) evidence, a second investigation was performed by a different evaluator. This person interviewed different team members and checked the findings again and in more detail. He focused explicitly on a study of all available documents.

*8) Implementation*

During this last step there was some involvement in the implementation of the suggested improvements to the design processes. This researcher's main role was that of a member of the product architecture team that was formed to develop a new product architecture, one of whose goals was to facilitate system-level coordination. Working together with the company was advantageous in increasing the understanding of product architecture in general terms, and it enabled the effects of the analysis to be observed. Ways of measuring the actual effects of the analysis on project improvement, and any difficulties involved in doing this, could also be investigated.

### *6.1.4 Summing up*

In this section, the case study objectives have been described and the case study setup was outlined. The data collection protocol consisted of a great number of steps, most of which were used in combination to reach the research objectives. Figure 6.4 provides an overview. The data collection steps are shown on the vertical axis, and the four research goals on the horizontal axis. It should be noted that the role the shaver itself has in reaching goals 2, 3, and 4 is placed between brackets. This is to indicate that the object provided no real data on the coordination required, but it did assist the collection of data.

| | | Research objectives | | | |
|---|---|---|---|---|---|
| | | 1 Validity Interactions | 2 Relation Coordination | 3 Understanding/ Improvement | 4 Project Performance |
| Data Collection | Documents | x | x | x | |
| | Shaver | x | (x) | (x) | (x) |
| | *Interviews:* | | | | |
| | Documentation interactions(3) | x | | | |
| | Identifying coordination (4) | x | x | | |
| | Presentations (5) | x | x | x | |
| | Key informants (6) | x | x | x | |
| | Second evaluation (7) | | x | x | |
| | *Implementation (8)* | | | | x |

▸ **Figure 6.4** How the objectives were established

To sum up, it is argued that the criteria for scientific quality as specified in Chapter 1 were met. The following constructs were satisfied:

- *Construct Validity*: The interaction constructs were firmly embedded in theory, multiple sources of evidence were used, and in particular a real-life physical product formed the basis of the research.
- *Internal Validity*: The internal logic of the taxonomy formed one of the foundations of the research's validity, with multiple units of analysis being used for literal and theoretical replication.
- *Reliability*: The way the interactions have been documented is highly reliable since it is based on a shaver readily available, and the results can be checked by anybody with sufficient technical knowledge. The reliability of the coordination is underpinned by the data collection protocol, and the fact that a second evaluator was brought in.
- *External validity*: The theory behind the taxonomy enables analytical generalization based on the broad context in which the underlying prescriptive literature is valid. However, for greater generalization of the study, an additional multi-case study is required.

The next sections set out the results of the study. Section 6.2 illustrates the interactions between the building blocks, section 6.3 describes the effects of each interaction on actual coordination, and section 6.4 reports on the analysis and formulation of options for improvement. Finally, section 6.5 explores the effects of the analysis on the performance of the company.

## 6.2 The interactions between the building blocks

In this section, the interactions between the current shaver building blocks will be illustrated and described for each of the three types of technical interaction. How these interactions are documented is based on the interviews described in section 6.1. For each interaction type the interactions are shown within a separate matrix. An interaction between building block A and building block B is plotted in the element (A, B). Since the interactions are symmetrical, it is sufficient to depict the lower triangular part of the matrix. The diagonal elements are plotted black since they obviously represent combinations of the same building block (e.g. A, A). It should be noted that this does not imply that there are no interactions *within* the building blocks. The internal interactions, however, are not within the scope of this research (as argued in Chapter 5).

For reasons of confidentiality, the building blocks of the shaver were randomly labeled from A till J. The representation and analysis remain fully valid, though, since the documented interactions were not changed.

### *6.2.1 The functional interactions*

The matrix below illustrates the functional interactions between the shaver's physical building blocks.

| Blocks | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| A | | | | | | | | | | |
| B | | | | | | | | | | |
| C | | 2 | | | | | | | | |
| D | | 2 | | | | | | | | |
| E | | 2 | | | | | | | | |
| F | | 1 | | | | | | | | |
| G | | | | | | 1 | | | | |
| H | | | | | | 1 | | | | |
| I | | | | | | | 3 | | | |
| J | | | | | | | 1 | | | |

*1 = Exchange of Energy*
*2 = Exchange of Material*
*3 = Exchange of Information*

▸ **Figure 6.5** The identified functional interactions between the building blocks

The studied shaver involves relatively few technical interactions (8 out of a possible 45) of this type. Number 3 in the matrix visualizes the exchange of information between building block C and building block H. Furthermore, between building blocks B, C, D, and E hair is exchanged because cut hair has to be transported and stored within the shaver. Of course, the hair is not necessarily input for the functioning of these building blocks. Finally, number 1 represents the functional exchange of energy between building blocks B, F, G, and J required to eventually perform the shaving function. These interactions indicate the flow of energy from the electric point (in the wall) to the shaving heads.

### *6.2.2 Mapping interactions*

Figure 6.6 depicts the mapping type of interactions by illustrating the function that is mapped across a pair of building blocks. In the event of a function being mapped across more than two building blocks, i.e. A, B and C, this function is depicted for all pairs involved: (A,B), (B,C), and (A,C). Note that the functions that are fulfilled by only one physical building block are not visible in this matrix.

In this analysis, styling is represented by the function 'look nice' and this function is represented by the symbol (☺). All of the physical building blocks fulfill this function jointly, except for building blocks B, C, F and G. Furthermore, four technical interactions occur because the building blocks E, F, H and I jointly protect the shaver against water (indicated by the symbol (🌢). Next, Building blocks B, C, D, and F engage in many technical interactions, because of the function 'follow contour' (〰) and 'remove waste' (🗑). A good shaving performance involves closely following the contours of the male face. In addition, the function 'cleaning' represents the opening and closing of the shaver to clean and remove the waste. Finally, the function 'protect against falling' (⊞) gives rise to 11 technical interactions between the building blocks.

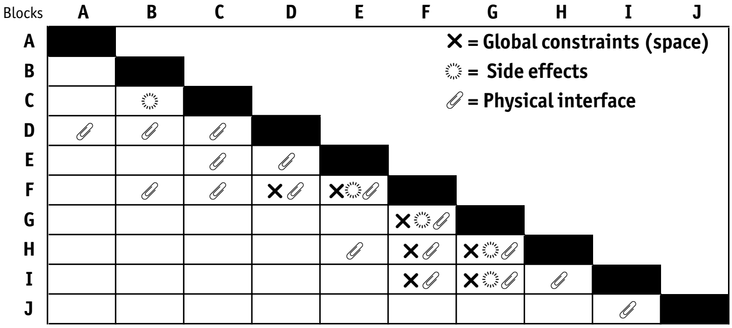
| Blocks | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| A | ■ | | | | | | | | | |
| B | | ■ | | | | | | | | |
| C | | 🗑 | ■ | | | | | | | |
| D | ☺⊞ | 〰⊞ | ☺〰🌢 | ■ | | | | | | |
| E | | | 〰 | ☺🌢⊞ | ■ | | | | | |
| F | | 〰🗑 | 〰🗑⊞ | 〰🗑⊞ | 🌢⊞ | ■ | | | | |
| G | | | | | | | ■ | | | |
| H | | | | ☺ | ☺ | | | ■ | | |
| I | | | | ☺ | ☺🌢⊞ | 🌢⊞ | | ☺🌢⊞ | ■ | |
| J | | | | | | | | ☺ | ☺ | ■ |

**☺ = Look nice (styling)**
**🌢 = Protect against water**
**〰 = Follow contour**
**🗑 = Remove waste**
**⊞ = Protect against falling**

▸ **Figure 6.6** The identified mapping interactions between the building blocks

### *6.2.3 The physical interactions*

Figure 6.7 illustrates the physical interactions that take place between the building blocks. Each symbol refers to the specific sub-type: global constraints, side effects, and physical interface. Contrary to the previous two types of technical interactions, building block G interacts with building blocks F, H, and I (see symbol ✕) because they share a limited amount of space. Side effects then cause technical interactions between building blocks F and G and between building blocks G, H and I. These technical interactions are represented by the symbol (◌) and in this case represent electro-magnetic radiation, heat and vibration. Finally, the physical interface of, for example, building blocks F and G is shown by the symbol (📎). We see that the physical interface is an important cause (15 times) of technical interactions; building block F in particular is a key player, interacting with 6 other building blocks.




▸ **Figure 6.7** The identified physical interactions between the building blocks

### *6.2.4 Final remarks*

On the basis of the data collection, the three proposed types of interactions between the building blocks could be clearly identified. The four different evaluators documented the interactions remarkably similarly. There were no different opinions about the documented interactions. The only differences occurred with respect to the side effects. For some pairs of building blocks only three of the four observations included a specific side effect. After comparison of all of the results, some designers mentioned that they had 'overlooked' a side effect. Everybody eventually agreed on a final set of interactions.

Summing up, many different interactions of all types were identified. The electric shaver involves relatively few functional interactions and most of these refer to flows of energy. There are quite a number of functions that are mapped across multiple building blocks. As such, a shaver clearly deviates from a fully modular one-to-one mapping model. It is striking that building block G does not engage in any mapping type of interaction. This fact will be brought up in the discussion. Finally, there are many physical interfaces between the building blocks. Contrary to the mapping type, building block G can be seen as a champion of side effects and global constraints.

## 6.3 System-level coordination

Having illustrated the shaver's interactions in the previous section, now is the moment to link these to the actual coordination required during the design process. As described in the data collection protocol, team members were asked to identify the coordination required in terms of the shaver's product architecture. This has been done for each interaction between each pair of building blocks. The results for each type of interaction will be described separately and some general comments will be made.

### *6.3.1 System-level coordination for functional type interactions*

Functional interactions involved a relatively small amount of coordination and it was relatively uncomplicated. During the starting phase of the project, a few experienced designers made a great effort to identify a set of functional specifications. These could be easily described and were informally communicated to all of the design teams. From that moment there was no need for the teams to discuss this topic any further. All teams focused on their individual solutions, bearing in mind that the teams would jointly supply the appropriate specifications. However, at a late moment during the design process, one team was unable to make their functional conditions work. As a result, new and ad-hoc system-level coordination was carried out to establish a new set of functional conditions. This obviously meant that each design team had to make a good number of changes to the decisions that had already been made.

*6.3.2 System-level coordination for mapping type interactions*

Interactions of the mapping type clearly indicated a need for intense system-level coordination. In order to make a shared function operational, the design teams involved had to collaboratively solve problems. In all cases it took a long time before the conditions for each team were clear and they could proceed with their work concurrently. In fact, the teams frequently had to make mutual adjustments to their detailed design decisions due to the mapping interactions.

For the mapped functions 'look nice' and 'protect against water' two overarching teams were temporarily created in addition to the individual design teams. The first overarching team mainly consisted of members of the industrial design department, and the second included experienced designers from mechanical design, production and engineering. The main reason for the creation of these teams was the 'newness' of these functions for the company. The styling of the shaver obviously had to differ substantially from previous models. Furthermore, the waterproof concept was completely new to the firm. As a result, different technical concepts were generated and tested before the main solution was chosen. For both interactions weekly meetings were instituted in order to coordinate the design teams.

For the other mapping type interactions there were no formal coordination mechanisms and coordination was achieved in a 'lateral' fashion by mutual adjustment. Note that there was much more experience in performing the technical aspects of these functions. This will be dealt with in greater detail below.

For the mapped function 'protect against water' and for the styling of the shaver many alternative solutions were reviewed. The chosen way of generating the ultimate solution had a generate-test character, and whenever it needed to be tested, this had a considerable impact on the building blocks involved. As a result, during this period the design teams were unable to make detailed design decisions in isolation of each other. The teams all relied on mutual adjustment to establish the system-level coordination and to adapt to the changes made necessary because of the evolving mapped solution.

The 'look nice' function had a particular impact on the design teams due to the intrinsically fuzzy character of 'nice'. To find out whether a specific styling is sufficiently 'nice', the proposed concept had to be tested by a worldwide panel of representative customers. Many different external changes were proposed, but for each test it was hard to predict if or which one would pass the test. The timing of the tests was planned, but their outcome and possible need for an additional test (with new styling proposals) could not be planned for. This obviously had a considerable effect on the design teams.

The final solutions were found relatively late during the project, but from that moment a feasible set of parameters could be defined for each building block, and these were communicated to the design teams by CAD/CAM. From that moment the teams could focus on their design tasks in relative independence of each other.

For the other mapping interactions it was roughly the same story, even though the teams had much more experience with finding the basic solutions for the mapped function.

The function 'protect against falling' affected many blocks during the design process. A test for whether the blocks could jointly fulfil this function correctly could only be performed at the end of the design process. Simulation studies to predict the effectiveness of the solutions were not yet possible since there were too many detailed design parameters involved and in too complex a fashion. During the design process, designers in the design teams adjusted their settings a number of times on the basis of experience and general design rules.

The blocks C, D, E and I, J involved a multiple number of mapping type interactions. This required considerable mutual adjustment across the design teams during most of the design project. Engineers stressed the complexity and non-standard character of these joint multiple interactions and mentioned the difficulty of dividing the work between multiple designers. One said:

> *While I prefer to assign one engineer to the task of designing both blocks, it is unfortunately too much work for one person. Several people are now doing these tasks, spending half their time coordinating.*

Another one stated:

> *Speeding up the design of these chunks is hardly possible: I'm working as hard as I can, and increasing the number of people for this design task will only slow down the process because we would have to coordinate all day.*

The interactions between blocks B and F required intense coordination within the one room. These blocks engaged in two interactions of the functional type. In this case not only the coordination of each interaction separately was difficult, but the interplay between the two mapping types caused enormous coordination problems. One designer explained why:

> *If I make a small change to block B this affects the shared function 'contour following' such that a change has to be made to block F. However, if an appropriate adjustment is made to F this also affects the shared function 'clean waste' and thus I need to change block B again. Hence, and I think you will already have guessed what I'm going to say,, this in turn changes the contour following...!*

As a result this process required a great deal of trial and error and the lead engineer had to pay it a lot of personal attention. Obviously, once a solution is found, these settings are particularly sensitive to small changes.

Unlike all the other building blocks, G engages in no interactions of the mapping type, as was previously mentioned in section 6.1. Strangely enough, the design team of G preferred to work within their functional department instead of the shared one room. They argued that there was no need for intense coordination with the other design teams, though for team-spirit purposes they would stay together.

Finally, many engineers stressed how sensitive the mapping interaction specifications were to small late changes. They mentioned that on a significant number of occasions, unpredicted changes occurred during a late stage of the design process, which in turn caused a considerable amount of system-level coordination to identify a new set of overarching specifications.

### *6.3.3 System-level coordination for the physical type of interactions*

In general, the physical interactions were fully specified until late in the design process. The coordination history of each sub-type will be treated separately: global constraints, interface, and side effects.

To start with the global constraint interactions: it was previously observed that building block G was involved in a number of global constraint interactions due to a limited amount of space. The coordination of these interactions was closely related to the styling function. During the styling 'selection period' the spatial interaction was subject to many changes and required a frequent update with the neighboring design teams. For each alternative style that was proposed, the design teams had to adjust mutually in respect of an appropriate and realizable division of space. These discussions took a long time and involved much trial and error at the level of fine detail since each proposed style imposed a lot of constraints on the available space. From the moment the final styling was determined, the available space was divided (by the lead engineer) and allocated to the blocks involved. The constraints could easily be communicated to all designers via CAD/CAM. For most design teams very little system-level coordination was required afterwards. Unfortunately though, the space constraint for block G turned out to be extremely tight. This resulted in a significant number of redesigns of block G in order to find a satisfying physical solution within the given spatial boundaries. While these redesigns did not themselves require any additional coordination between the teams, in a few cases the team had to beg a neighboring team for slightly more space, which obviously went together with additional system-level coordination. However, unless the interactions were constantly in the minds of the G design team the amount of system-level coordination was relatively low, and all teams could generally perform their tasks in relative independence.

Second, the coordination with respect to the physical interface differed per interaction. In the majority of the situations, the interfaces were explicitly considered after the other types of interactions were clearly known. A design involving an interface between two building blocks was often performed by one of the two teams and required little system-level coordination. The electric wire between the drive unit and the power unit was, for instance, seen as a 'distinct' component that could be performed by one team. However, designers mentioned high levels of coordination when it was not clear which team was responsible for the design, or when the proposed interface was strongly biased toward the design team in charge, and unacceptable for the other block involved. Furthermore, late adjustments to the interfaces involved intense trial and error between the teams since most design parameters were already fixed by other interactions and accordingly the 'solution space' was extremely limited. The lead engineer paid great attention to these interfaces and took care of the coordination. It was nevertheless considered much too much work for one person.

Third, the coordination of the side effects had an ad-hoc character. From the moment a side effect was detected, emergency meetings or temporary teams were created (initiated by the development management team). The temporary teams considered proposed changes and set engineering change protocols in motion. Accordingly the side effects required a great amount of mutual adjustment and close face-to-face contact since the matter was very complex and sensitive to small changes. Furthermore, the effect of proposed changes could only be tested after these had been fully implemented. In some cases the side effects could be solved by small changes within one building block, in other cases trial and error between multiple blocks was necessary. Things became even more complex when proposed changes affected design parameters that were also part of other interactions, and produced a need for an additional amount of system-level coordination. For instance, increasing the distance between two blocks was able to solve problems related to excess heat, but this was strongly constrained because of the chosen styling.

### *6.3.4 Summing up and overall remarks*

To sum up, the coordination required could be linked to each interaction. The functional interactions and global constraints required relatively little system-level coordination effort. Once these specifications were set the teams could do the detailed work in isolation of each other. In a few situations this was hampered when one block failed to meet its initial specifications and this gave rise to a need for additional system-level coordination.

The mapping interactions generally went with intense mutual adjustment between the teams involved. Working in isolation of each other with respect to these interactions turned out to be a pleasant but non-realistic perspective. During most of the design process, the design teams needed to coordinate detailed design decisions. In the first place it took a long time before the specifications of a shared function were collaboratively defined. Second, these specifications turned out to be sensitive to small late changes, and thus created a need for additional system-level coordination.

The physical interfaces turned out to be an important issue for system-level coordination during the latter parts of the design process. In general, each interface was designed by one team, which obviously required very little system-level coordination. However, the most important reason for coordination being needed was the initial decision of which of the two teams was responsible for their physical interface, and the subsequent testing of the interface.

Finally, the side effects required a considerable amount of system-level mutual adjustment. The design teams had to discuss each other's detailed design decisions and adjust and test their parameters collaboratively. However, the specific features of coordination strongly depended on what technical issues these side effects referred to.

## 6.4 Towards options for improvement

In the previous two sections design team system-level coordination was described in terms of the underlying architecture of the product. In this section, how this resulted in the design teams reaching a common understanding and the generation of options for improvement for future design projects will be described. The general findings will be introduced below, and the form in which the results will be presented here will be explained in greater detail.

### *6.4.1 Introduction*

At the time of this analysis the design team had almost finished their design project and the entire process was fresh in their minds. The project had had a significant impact on the firm and had not gone unnoticed by any of the groups within the plant. There were mixed feelings about the project. On the one hand the team was really proud of having pushed themselves to the limit to bring a new and innovative shaver to the marketplace right on time. On the other hand the amount of effort that had gone into it was tremendous. The project team was continuously forced to search for non-standard solutions and needed to spend enormous effort on coordination. Various parties within the organization were searching for explanations for this intense effort and were looking for ways to prevent these situations occurring in the future.

The analysis thus came at the right moment in time. It enabled a clear overview of the established interactions between the shaver building blocks and emphasized the importance of system-level coordination. This idea was not a familiar one within the company and resulted in reflection about the product architecture, and the way they managed the interactions in general was put under scrutiny.

The need for this could be explained, and options for improvement per identified interaction could be suggested. How each interaction was embedded within the broader contingencies of the firm could also be clearly deduced.

The analysis was presented to groups of designers, project leaders, and management and it became clear that these parties saw the importance of different aspects. Broadly speaking, the project leaders were concerned with the organizational structure, planning, and allocation of responsibilities during the project. The designers took a very broad view, but highlighted the importance of day-to-day coordination of the interactions in particular and discussed the causes for these interactions. In their turn, the management team generalized the findings of this project to all ongoing and future development projects and focused on the need for a change of architecture. Their basic reaction was that some 'major' interactions needed to be removed in future projects. Accordingly they focussed the discussion on major contingencies (cost, shape, etc.) and priorities that affected the major technical interactions involved in the product.

It should, however, be said that this is not to imply that the designers did not consider strategic matters or that management was not concerned with the operational aspects; these are merely the broad differences. The findings will be structured in the following fashion.

- The company's ideas about how architecture and system-level coordination is generally managed will be described.
- How well the required coordination can be deduced and what options for improvement were suggested will be explored.
- How the interaction structure is embedded in the broader contingencies of the firm will be highlighted.

### *6.4.2 Improvement in general awareness of system-level interactions*

Representation of the architecture of the new shaver created an awareness of the occurrence of some interactions that had not previously been noticed by all parties. There had been a belief that the shaver was composed of independent modular building blocks. This is actually not so surprising.

At a general level, the interactions gave a structured unambiguous overview of the great need for system-level coordination during the design project. This triggered discussion and reflection on the organizational structure of the design projects and what aspects could have been done better. The main issues are described below.

The project team was of the opinion that the current match between the product and the design teams was the best one possible. Despite the members now realizing the considerable number of interactions between the blocks, they argued that there were considerably more interactions *within* the blocks, hence a change in the structure of the design teams would obviously result in even greater need for system-level coordination.

The one-room approach was considered very beneficial since frequent lateral exchange of information between the design teams was needed to manage the interactions. The team mentioned, however, that the one-room approach went at the expense of learning from other teams and functional departments. Hence, for block G (the one with no mapping interactions) it was suggested that this take place (or partly so) within their functional department. For the others, though, coordination of the workflow was given the highest priority.

The design team suggested that the team's weakness was the underestimated importance of structural and formal system-level coordination. The easy exchange of information between the teams did not force the project members to handle the interactions in a structured and deliberate way.

Formally speaking, the lead engineer was responsible for the management of all system-level interactions. However, given the great number of interactions it became clear that one person could not possibly handle and control all these interactions. This management overload, however, was not formally compensated for at the design team level.

Furthermore, the meetings that were instituted to handle the interactions between the design teams were considered to involve too many people. In these meetings all of the interactions between all of the building blocks were addressed. Since engineering issues tended to involve only a few interactions per engineer, they tended to lose their concentration.

Another issue that was mentioned was that according to the designers, the work was divided into too many tasks. For the design project in this analysis, the number of team members was significantly larger than for other design projects. The major reason for this was to increase the speed of the process, since no time was lost because of shortage of designers. As a result, the same amount of work was divided across more people and the design tasks became much smaller. This meant that much more coordination was required within the design teams and there was less focus on matters outside the design team. It was, however, striking that some designers helped other teams when they had finished their initial tasks, though this was usually to perform a small and local task. Somewhat paradoxically, switching designers between the teams did not increase understanding of the need for system-level coordination, but rather reduced it.

To sum up, the focus on the interactions between the building blocks made the emphasis on the issue of system-level coordination one of major importance. This connection had not previously been made explicit, and it initiated a discussion on how system-level interactions were managed in general. Despite the generally recognized advantages of a design-team structure, the system-level interactions could have been managed much better. The focus on the teams was greater than the focus on the interactions. In effect, the following measures for improvement were suggested:

- In addition to defining the role of the lead engineer, clearly defining responsibilities in relation to interactions between the design teams.
- Initiating smaller and more focused meetings to formally handle the interactions.
- Preventing the tasks from becoming too small in order to increase the focus on system-level interactions.

These options for improvement are a direct result of the emphasis on interactions. It should be noted, however, that the last recommendation in particular has little to do with the theories described above. This point will be brought up again in the discussion. In the next section, use of the analysis to explain and improve the individual interactions will be described.

### *6.4.3 Improvement per interaction*

Since the documented interactions were clearly categorized according to the proposed taxonomy, the inevitable consequences for system-level coordination could be explained. Again, the team members fully understood the natural consequences of coordination embedded in a specific type of interaction. Translating this 'logically' into system-level coordination for each interaction had clearly helped a lot as it reflected the way the design team actually worked.

Some team members (the designer, for example) tended to make comments such as the following:

> *Now you can see that it is not our fault that coordination between our two design teams is taking so long; it is due to that mapping type of interaction. If we want to speed things up we really have to do something about that interaction..*

On the other hand, some retreated to the security of their team and commented that in some cases, considerably more effort was spend on coordination than in retrospect seemed necessary for a particular interaction. This was one comment:

> *Maybe we could have organized that functional interaction much more effectively than we actually did. Why were we working that way and how can we improve in the future?*

Below is described what options for improvements were suggested for each type of interaction. These options contain both suggestions for manipulation of the interaction and suggestions for improved coordination. The traditional sequence will be followed: functional, mapping, and physical interactions.

#### *Improvement of functional interactions*

Things went smoothly during most functional interactions. However, the engineers have now recognized that the functional interactions were only suitable for working concurrently under the condition that the functional scheme is clear and realizable. Accordingly, a number of suggestions were proposed.

- Focus on functional scheme.

Most designers had years of experience in designing shavers and were used to thinking in terms of physical solutions. The construction of a functional scheme did not have the highest priority since in their view the main solutions were already known to a large extent. They did not expect that a functional scheme could add much. However, more emphasis on a functional scheme could have had many advantages with respect to system-level coordination. The scheme could have been finished earlier and have been presented in a more formal and structured manner so that the design teams could have started their parallel work at an earlier stage of the design process.

- Risk management.

Since the teams could work in isolation of each other (for functional interactions) they lost track of the progress of the other teams and were not aware of the gradually increasing risk that one of the teams would not manage to fulfil its functional specifications. As a result most teams were confronted with unexpectedly having to reformulate functional specifications at a late state of the design process. If this risk had been identified at an earlier stage of the design process, a lot of coordination and effort within the teams could have been spared. It made it clear that each team has a responsibility to realistically estimate and communicate the likelihood of failure. Some team members mentioned that avoiding a team attitude of 'failure is not an option', or 'we'll do everything we can to solve the problem before we bother the rest of the organization' would be to the benefit of transparent overall coordination. This would suggest that the risk of a team not fulfilling its functional specifications at the end of the design process should be explicitly addressed and managed. Successful risk management depends on two parties. The design teams must be open about what progress is being made. On the other hand, (project) management needs to recognize and control the risks.

- Enlarge the range of functional specifications.

With respect to the 'hardware' of the interactions, system-level coordination would have been improved by the functional specifications having more 'slack'. For instance, coordination becomes easier if the power unit is designed to supply electricity within the range of 4 to 5 volts, although a range of 4.5 to 4.6 volts would suffice. Functional over-specification reduces the time that is needed to construct a realizable functional scheme and therefore facilitates working concurrently. Moreover, 'slack' also reduces the risk of failure by the team itself, or reduces its sensitivity to the failure of other design teams, and accordingly diminishes the probability of additional coordination being needed. This option for improvement would clearly have a considerable impact on the unit cost price of the shaver, and this would have to be taken into account later on.

- Predevelopment.

For those building blocks that failed to formulate their functional specifications, predevelopment would have been a good option. Construction of the overall functional scheme would obviously have been faster if research had been done in advance of the project. In addition the probability of failure would have been considerably reduced if there had been more readily available knowledge of the physical solutions.

- Standardization.

Similarly, an existing solution obviously needs the least system-level coordination involved.

*Improvement of mapping interactions*

With the mapping type of interaction, the underlying nature of the interaction may explain what coordination is required. As such, the most effective way of bringing about improvement is to change the mapping interactions and thus reduce the need for coordination. Improving the coordination is another option. Some suggestions for doing so are the following:

With respect to this coordination, a project member mentioned that once a solution has been found for a particular shared function (whether this is done collaboratively or not) it is important that the conditions remain 'sacrosanct' for the rest of the project. The reason for this is that every time the conditions are violated new system-level coordination is required to identify new solutions, and this will effect the other design teams. This provoked the following suggestions.

- Check and manage solved mapping interactions.

In this regard, the project team mentioned several cases where an already solved mapping interaction was 'disturbed' by accident. The geometric solutions of the mapping interactions were documented and communicated within a CAD system. It turned out that such a system is not completely suitable for communicating the exact agreement since too many things may be affected. Inexperienced designers made small changes to 'their' blocks without the system warning them that they were affecting a mapping interaction. It is thus advisable that already solved interactions be continuously checked and managed by an experienced designer. The next remark is similar.

- Avoid unnecessary changes.

Making changes to building blocks that unnecessarily affect mapping interactions should be avoided. Designers mentioned that some late changes within a building block unnecessarily affected a shared function and consequently initiated additional system-level coordination. One reason for this was that team members became used to dealing with changes since mapping interactions take a long time to coordinate informally. Hence, when changes were proposed at a late stage of the design, there was no system for alerting the teams that this could be a dangerous move. Instead, the teams rolled up their sleeves and set to work. Likewise, it was not always clear when the design teams were still looking for a feasible set of design parameters for the mapped function, or when the conditions for each building block could be considered set and fixed. To sum up, the unintended drawback of high levels of mutual adjustment is perhaps that it does not force designers to draw clear boundaries around decisions that have already been made. It should be noted that while unnecessary changes should obviously always be avoided, no matter what type of interaction, the mapping type is particularly sensitive, and the consequences for coordination are especially harmful.

- Predevelopment.

As with functional interactions, prior knowledge facilitates coordination. For the function 'protect against water' this would probably have prevented the need for much system-level coordination. Though predevelopment of a mapped function can be more complicated than for the functional, since it is much more influenced by the detailed characteristics of at least two of the blocks, it should be noted that predevelopment of the 'look nice' function is hardly an option since it is too dependent on fashion.

As mentioned before, these types of interactions are intrinsically difficult to coordinate, and it must be said that the above remarks notwithstanding, the design teams did a good job. As such, the technical manipulations that the interactions involved would seem to be preferable to setting up a system to facilitate system-level coordination.

- Reduce the number of mapping interactions.

To take an example, if the function 'protect against water' could be fulfilled by one new building block, then the number of mapping type interactions would be significantly reduced. All coordination with respect to that function would take place within the building block, and no system-level coordination would be required at all. The same applies for the other mapping functions.

- Simplify mapping interactions.

It was found that for the mapping interaction between B and F, additional conditions were included for marketing reasons. Further investigation showed that these conditions could easily be relocated to a different internal building block location, and as a result greatly simplified mapping interactions and the related need for coordination.

· Standardize a mapped function.
Standardization has the benefit that the detailed specifications of the mapped function are set in advance and no system-level coordination between the design teams is likely to be required. The drawback, though, is that such a standard interaction greatly restricts the technical decisions that then have to be made within the design team.

It should be noted that all options that affect the physical product seem to be quite effective in reducing the need for system-level coordination, but are also strongly constrained by available technology, marketing and other contingencies. This aspect will be examined in greater detail in 6.4.4.

*Improvement of physical interactions*
The physical interactions will be addressed per subtype

– *Global constraint*
With respect to global constraint interactions, it was observed that it took a relatively long time before the available space could be divided across the several building blocks, but that afterwards parallel development was possible, though for block G the constraints were really tight. The following options are thus available.
· Make the division of space as realistic as possible.
According to the design team, the allocation of space could have been done with more care and more regard to feasibility.
· Relax the global constraint
The coordination effort that is involved with a global constraint can be reduced by creating more space. This speeds up the decomposition process and reduces the probability of failure. Two options come to mind. The first is for styling to take the interior dimensions of the building blocks more into account. The shaver would probably become slightly larger or, alternatively, altered at some at some points. The second alternative is to miniaturize the building blocks by choosing other technical solutions. The choice of a smaller motor (with the same functionality) would reduce the pressure on space and facilitate system-level coordination considerably.

– *Side effects*
The project team was well organized to handle side effects. Emergency meetings or temporary teams could be quickly instituted, and exchange of information between the designers involved was easy. However, the following options should be considered:
· Speeding up of the procedure for engineering change.
When changes have to be made to the original specifications these have to be agreed upon by all parties involved. By their very nature, these are relatively long and possibly hierarchical procedures. Some designers mentioned that speeding up the process may have benefits, and there were some suggestions made about how to do this. Looking at it in terms of Galbraith's theory, this would include the option of increasing the capacity to handle exceptions. In fact, faster change protocols would be beneficial for all interaction types, but are most relevant for this situation.
· Predict side effects.
When side effects can be predicted in advance, the coordination can be transformed to a

mechanism of goal-setting and much less system-level coordination will be required in the later stages of the design process, though as noted in the description of the results, for mechanical matters, this may be difficult to predict in advance.

- Prevent side effects

The most effective solution is to prevent side effect. To do so, particularly robust solutions with not much chance of side effects should be given preference. However, these are likely to entail additional costs. A more reactive solution would be to build in the possibility of being able to easily make changes if side effects do occur.

– *Interfaces*

As was previously mentioned, the best way of improving interfaces is to make clear decisions about who is responsible for their design. In addition, standardization of the interface would greatly facilitate coordination. However, the other interfaces and contingencies will heavily influenced the feasibility of standardization.

To sum up, as shown above, each type of interaction can be improved, and the best way of doing this will depend on the type of interaction. The focus will now shift to the underlying contingencies of the interactions and a broader light shed on the manipulation of interactions.

### *6.4.4 The contingencies behind the interactions*

In Chapter 4, the importance of understanding how an interaction is embedded within the organizational structure and way of working was described. Since the technical reasons behind the documented interactions were clear, it was possible to analyze the underlying reasons for specific technical interactions being chosen. Most technical choices could clearly be traced back to purposefully set project or company goals. With the shaver, the number of technical interactions between the building blocks could be explained by a design policy of optimizing styling and unit cost and maintaining the fit with the production structure.

#### *Low unit cost*

For the functional interactions in particular, 'over specification' was suggested as being the best way to facilitate coordination. However, the present 'narrow' specifications of the functional interactions can be explained by the ongoing pressure to reduce unit cost. Since the shaver is a high volume product, every saving of a cent in the dollar results in enormous overall cost savings.

A power supply with minimum specifications is cheaper than one with greater functionality. One designer explained this in the following way: 'Even those that don't work have to be feed'. If the company wants to reduce system-level coordination in the future (by altering the interaction) the unit cost represents the greatest barrier. The problem here (one recognized within the literature) is that while it is a straightforward matter to calculate the increase in unit cost, it is far more difficult to calculate the overall effect.

The cost factor is also the most pressing reason for not choosing the option of miniaturizing the building blocks in order to facilitate global constraint interactions. Smaller components are generally more expensive than larger components with the same functionality.

The need for the many interactions may also be related to cost considerations. For two of the mapped functions within the current design the functions can be carried out by the already available design parameters of the two blocks. Hence, if the function were allocated to one block, additional design parameters would have to be included in that block, which would in turn increase costs.

*Styling policy*
The styling policy also reduced the space available for block G and caused global constraint interactions. In fact the global constraints could have been managed more effectively if the inside building blocks had been designed first, and the styling determined afterwards. However, this was not in line with the overall priorities of the firm. Styling always takes first priority and the least restrictions possible are imposed in order to achieve maximum customer satisfaction. In the options for further research this issue will be looked at again.

*Assembly system*
The established assembly system works best with components of a specific size. This has resulted in the choice of relatively larger components than would have been chosen otherwise. In combination with the high standards for styling, the larger components in turn increase the global space interactions.

*Production structure*
The available production structure strongly influenced the technical choices that gave rise to the mapping interactions. The fact that there were four distinct production units restricted the technical options for the function 'protect against water'. This resulted in a great number of mapping interactions. Some options for manipulation were also hampered by the firm's existing form of mechanization. A change in interactions requires investments in new production technology, which in turn has an impact on costs.

To sum up, a significant number of technical interactions can be traced back to the firm's broader contingencies. These contingencies are deeply embedded in the firm cannot be changed overnight. When the firm makes the decision to speed up the design process by manipulating the technical interactions these go at the expense of the firm's traditional policy. As such, any wish to increase the performance of design projects has to be weighed up against the company's other contingencies, and new priorities need to be set.

## 6.5 The implications of the analysis

The foregoing sections have dealt with the analysis and options for improvement. This section will describe the post-analysis period: what happened next. Its effect on the management of system-level interactions within new projects will be briefly discussed. The most significant change – the institution of an architecture team – will also be looked at. Finally, the difficulty of measuring the effect of all of the options on the actual performance of new design projects will be discussed. A performance indicator will be proposed which hopefully will provide empirical evidence of improved project performance in the future.

### *6.5.1 The effects within the design teams*

After this researcher and members of the company had presented the results of the analysis, two posters advertising their implications were created. These were hung up in all of the departments in order to increase awareness of the product architecture and its considerable impact on the organization of the design processes. After the analysis, the focus on interactions in new design projects increased. These were coordinated more deliberately, and to the extent that this was possible, the designers tried to minimize the interactions. They were at least able to take the coordination perspective into account when selecting technical solutions for new shavers. The clearest example of the new emphasis was the introduction of a formal decision document where the interactions were now explicitly included. This document includes reports from all of the design teams on their building block decisions and a matrix clearly indicates when the decisions made by one unit will affect those made by the others. The most remarkable change, however, was the introduction of an architecture team.

### *6.5.2 The architecture team*

The architecture team was instituted immediately after the presentation to the development management team. It was a multi-disciplinary team with managers from product development, manufacturing, engineering, logistic, internal consultants, and the researcher himself. Other disciplines such as marketing and purchasing were planned to be added in a later period.

The goals of the architecture team were twofold. The first was to broadcast the new vision on architecture to the whole company. The second was an exploration of the options for adapting the architecture so that it was more efficient. This included manipulation of interactions within the broader perspective of the disciplines and contingencies.

The second aspect related to most of the team's activities, and the initial steps that were undertaken. These will be briefly outlined.

#### *Outlining the overall impact of architecture*

As a first step, the team members identified the main issues in their particular field and linked these to the product's architecture. The main business goals of the company were then formulated and the contribution to these goals that could be made by an adapted (more modular) product architecture was roughly estimated and compared to the estimated effects of the traditional architecture. As could have been deduced, next to product development, the most important stakeholders of a new architecture were the areas of production and engineering.

#### *Diversity study*

The subsequent step included a much more detailed study of the production and logistical situation. The increasing emphasis on variety over the years was traced and the achieving of variety within constraints was addressed. The ideal picture of a relatively small number of different building blocks and a great variety of end products (variants) was compared to the existing situation on the floor. It turned out that over the years there had been a decrease in uniformity of building blocks and components. The company was obviously fully aware of this, but now we were now able to explicitly link the level of uniformity to the interactions between the building blocks. As a matter of fact, the interactions between building blocks not only cause considerable need for system-level coordination, they also result in unique physical solutions, which in turn decreases the overall level of uniformity. An analysis of which interactions between the blocks could be identified as most hampering standardization was done, and this knowledge used to largely determine the new architecture.

*Trade-off*

As a direct consequence of the analysis of system-level coordination and the above steps which took the production situation into account, the trade-off for a new adapted architecture could be calculated more precisely. On the one hand it was expected that fewer system-level manipulations would result in better design project performance and greater uniformity on the production-floor, but on the other hand it was likely that all of the options would result in a higher unit cost per shaver. Various scenarios were formulated and their overall effects calculated. These calculations were very complex and surrounded by assumptions, but ultimately led to highly beneficial courses of action.

At the present moment this process is still underway, and for reasons of confidentiality a more detailed illustration of the plans and implementations cannot be given. There is no doubt, however, that the analysis has had a significant effect on the new design projects and can be expected to have a positive impact on project performance. The next section will address this aspect in more detail.

### *6.5.3 Measuring performance*

The final research question addressed the impact of the analysis on the performance (time, costs, quality) of the project. It is, however, impossible to prove that the design processes have led to or will lead to better performance.

In order to investigate the relationship between the implementation of improvements and increased project performance at least two variables are required: an indication of the improvements implemented, and a measurement of actual performance.

The latter variable was particularly difficult to obtain at the time of this investigation. The major reason for this is that the throughput time of the new projects (after the analysis) went beyond the period allotted to this research project. Hence it is only possible to measure the overall performance of the new project in the longer term. Another way of viewing this is that the initial stages of the project cannot provide reliable information in the short term. A glance at the previous design projects will show that the initial performance of a design project within its first phases is no predictor of the final outcome of the project as a whole. In some cases high speed in the initial phases was compensated by relatively low speed at the end of the project. For other projects it was just the other way around.

In effect the only thing that can be done is try to find a way of measuring how interactions are handled within new projects such that these can be related to the performance of design projects in the long term.

Any assessment of how interactions are managed will be based on the fact that improvement will ultimately result in earlier full specification of the interactions between building blocks. The described suggestions for better coordination of existing interactions all highlight the importance of early and formal specification (where possible) of the interactions. In addition, it can be expected that all proposals for the manipulation of interactions will also stimulate this effect. In fact, the reduction of interactions, the simplification of interactions, and standardization of interactions will all make the specification of the interactions easier and facilitate early specification.

A performance indicator that focuses on specification of interactions between building blocks during each new design project will now be proposed. After each stage is completed, all interactions, whether functional, mapping, or physical, are fully specified for each pair of building blocks. This can be done on the basis of posing a standard question and putting it to each building block designer:

*Are the interactions with the other building block specified such these are completely clear and sufficient for an (imaginary) other company that has to design the other building block?*

The answer can be Yes or No. The results are shown within a matrix as depicted in the figure below.

| Blocks | A | B | C | D | E | F | G | H | I | J | # to be specified | # specified |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | - | x | x | 0 | x | x | x | x | x | x | 0 | 0 |
| B | - | - | 0 | 0 | 1 | 1 | x | x | x | x | 4 | 2 |
| C | - | - | - | 1 | 0 | 0 | x | x | x | x | 3 | 1 |
| D | - | - | - | - | 0 | 0 | x | 0 | 0 | x | 4 | 0 |
| E | - | - | - | - | - | 1 | x | 0 | 0 | x | 3 | 1 |
| F | - | - | - | - | - | - | 0 | 0 | 1 | x | 0 | 1 |
| G | - | - | - | - | - | - | - | 0 | 0 | 0 | 3 | 0 |
| H | - | - | - | - | - | - | - | - | 0 | 0 | 2 | 0 |
| I | - | - | - | - | - | - | - | - | - | 0 | 1 | 0 |
| J | - | - | - | - | - | - | - | - | - | - | 0 | 0 |
| | | | | | | | | | | SUM | 20 | 5 |
| | | | | | | | | | | % specified interactions | | 25% |

*Legenda* *0* *Interactions not fully specified*

*1* *Interactions fully specified*

*x* *This pair of blocks has no interaction*

▶ **Figure 6.8** An example of the proposed performance indicator

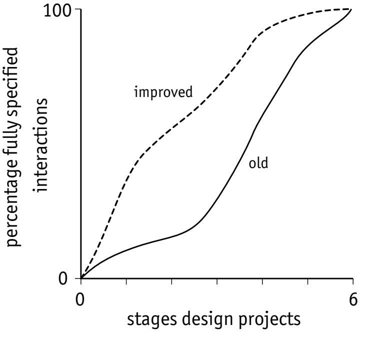
The matrix shows whether the interactions for each pair are fully specified or not, and indicates the overall percentage of specified interactions. These percentages can be compared during the evolution of the design process. This is shown in the next graph (Figure 6.9).

As a direct consequence of the analysis and implemented improvements, the percentage of fully specified interactions can be expected to increase. This is to say that at any specified moment of the design process, the 'improved' processes will have relatively more fully specified interactions than the traditional processes.

These percentages can be compared to the actual performances of the design projects under study. To what extent the options for improvements have been implemented can thus be determined, and related to performance. In the discussion to this aspect will be looked at again.

In addition to the measuring of performance, this representation can also be used by company management to check to what extent the recommendations of the analysis can actually be applied to design projects.

To sum up, it is not possible to indicate precisely how the analysis has resulted in actual improvement to system-level coordination and how these affect the performance of the projects. What can be said, however, is that the company has initiated many courses of action based on the analysis, of which the institution of the architecture team is the most radical one. Furthermore a way of modeling how all these actions result in better system-level coordination which in turn can be related to project performance has been described.




▸ **Figure 6.9** Comparing the percentages of fully specified interactions

*6.5.4 Summing up*

The representation of product architecture enabled system-level coordination problems within the electric shaver design processes to be clearly understood. This included the company gaining an increased awareness of the management of interactions in general. The current way of working could be seen and the project team suggested ways in which the interactions could have been handled better.

The second result is that each interaction could be discussed separately and coordination improved. Based on the characteristics of each type, better ways for coordination could have been suggested or technical changes to the shaver proposed in order to facilitate coordination.

The analysis also had the effect of enabling the overall company or product policy to be translated into interaction structures. This aided understanding that some interactions were unavoidable and showed the broad implications of interactions being manipulated.

## 6.6 Discussion of the results

In this section the results of the case study will be discussed and the methodological aspects introduced in the beginning of this chapter considered. This will be done for all four objectives of the case study. The same sequence in which the results were described will be adhered to.

First, the validity of the taxonomy for representing product architecture will be discussed. Second, the relationship between the interactions and coordination required will be described, and whether the findings meet the theoretical propositions or where these may be revised or extended . Third, the functioning of the tool in practice will be described. The logic of the taxonomy's ability to improve a design process will not be addressed in any depth, but the role of the analysis in general. Methodological considerations will also be addressed. Fourth, the problem of how to measure the actual effect of the analysis on project performance will be addressed.

These will all fall under the following headings:

- Discussion of the validity of the interaction constructs.
- Discussion of the relationship between the interaction types and the required coordination.
- Discussion of how to generate options for improvement.
- Methodological considerations.
- Discussion of the measuring of effect on improvements.

### *6.6.1 The validity of the interaction constructs: discussion*

During the thesis, the necessity for the interaction constructs to have high validity has been frequently stressed. In this researcher's view, a proper analysis of a product's architecture has to be based on this. To that end, considerable attention has been paid to producing clear theoretical definitions of the interaction types and the case study protocol designed in such a way that the interactions could be observed on frequent occasions.

The results of the case study strongly suggest that the three proposed types of interactions are valid. In the first place, per interaction, the four interviewees documented the interactions almost identically, independent of their background or moment of interviewing. Second, during the subsequent steps of the protocol there was no disagreement about the identified interactions, nor was it commented that there were missing interactions that had not been included in the interaction constructs. In all cases the interactions could be clearly linked to underlying technical reasons, which made different interpretations almost impossible.

The reliability of the interactions is considerable since these are based on a real-life product and can be checked at any time and independently of this investigation.

The taxonomy can be expected to be externally valid as far as representing the product architecture of a broad range of other products is concerned. The evidence for generalization is not to be found in the shaver case study, but is based upon the logic of the applied engineering design constructs (analytical generalization). Application of the taxonomy to other situations is obviously required for proof.

The high validity of the proposed taxonomy is extremely beneficial for the additional steps in the research. In this researcher's opinion, the taxonomy potentially provides a strong basis, theoretical and otherwise, for design project research and product architecture in general. That the taxonomy of interactions can be applied in a precise and reliable manner is considered to be one of the research's achievements.

### *6.6.2 Discussion of the relationship between the interaction types and required coordination*

In addition to high validity for the interactions, developing the interaction types such that these could be linked to coordination characteristics during the design process was aimed for. The goal was twofold: to make a first step towards a theory linking product architecture to organizational consequences, and to show that this would be helpful for understanding and improving system-level coordination during an actual design process. In this section the results with respect to the theoretical considerations will be examined. The second aspect will be discussed in the subsequent section.

In the case study, each documented interaction could usually be shown in terms of the system-level coordination required by the design process. This enabled all these observations to be compared or contrasted in order to identify evidence for a relationship between the interaction types and system-level coordination characteristics. There was a particular interest in the extent to which the proposed characteristics of each type of interaction match actual system-level coordination. In addition, some factors that had an impact on coordination effort, but were not directly included in our taxonomy, were discussed. The objective targeted in this discussion is the creation of a theory (or a first step in that direction) on the basis of which a particular type of interaction between building blocks can be matched with a particular type of coordination.

The individual findings in respect of each type of interaction will be discussed below in the following order: mapping, functional and global constraint, and side effects and physical interface interactions. Furthermore, the effects of the taxonomy's limitations (as outlined in 5.4) will be discussed. To that end observed interference between the interactions will be addressed, and time and cost issues briefly reviewed.

*Mapping*

The findings in respect of mapping types of interactions were remarkable in some respects. In all cases where these interactions occurred, considerable need for mutual adjustment between the involved teams was observed. This strongly corresponds to the theoretical expectations (literal replication). Moreover, the only team that was not involved with mapping interactions demonstrated a relatively low need for coordination. This result contrasts to those of the other design teams, but for a predictable reason (absence of the mapping interaction). In this researcher's view, this is an example of theoretical replication (see 6.1). These replications substantiate the proposition that a mapping type of interaction is involved with a considerable need for mutual adjustment during the design process.

*Functional and global constraints*

The functional and the global constraint interactions will be discussed together since these turned out to be quite similar. For the functional and global constraint interactions, a relatively low need for system-level coordination was demonstrated. Almost all teams reached agreement on the planned specifications of the function or constraint and then performed the required detailed design work concurrently and without any need for additional system-level coordination. This matched the theoretical expectations (literal replication). However, not all functional interactions could be matched with low system-level coordination. In one case, considerably more coordination was needed than theoretically expected following the failure of one team to meet its planned specifications. Furthermore, it became clear that where case constraints are very tight it may take a long time before planned specifications for each team can be formulated.

The taxonomy enabled these aspects to be analyzed. However, a future and more refined taxonomy could perhaps consider including factors such as 'probability of failure' or 'slack': had this been done in this research, the observations made at this point could have been more precise.

It should be noted that due to the limited impact of functional interactions on coordination, replication logic (comparing the coordination effort of blocks with and without functional interactions) was not effective.

To sum up, the results show that functional and global constraint interactions require a *minimal* amount of system-level coordination between the building blocks. Relatively little mutual adjustment is required and goal setting is an appropriate coordination mechanism.

*The interface and side effects*

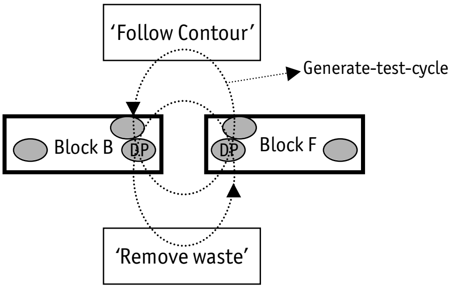
The relationship between the remaining physical interactions (side effects and physical interface) and actual system-level coordination is less straightforward.

In all cases side effects caused a temporary (unexpected) need for mutual adjustment between the teams. However the coordination needs differed across the interactions, and that literal replication could not be noticed. This could be expected since these by definition include some unpredictability, and may refer to all kinds of technical 'problems'.

For the physical interfaces we were not able to pose clear theoretical statements about corresponding coordination. In respect of the shaver case, the practical findings showed that physical interfaces involved mutual adjustment late in the design process. In most of the situations the interface design between two blocks took place within one (of the two involved) design teams. When properly managed relatively little coordination was involved. However, there were many reasons for additional coordination being required: unclear responsibilities, difficulty in meeting the wishes of both blocks, technical difficulties, the relationship to other interactions. In short, whether a physical interface interaction requires a specific type of coordination remains undecided.

*Interference between the interactions*

The coordination characteristics were discussed for each type separately above. During the case study however, unexpected interference between multiple mapping interactions between two blocks (e.g. B and F) was observed. The two interactions each involved a generate-test cycle between the design teams, but in addition these cycles affected each other. This is depicted in Figure 6.10. This phenomenon could not be explained according these types of interaction alone. A closer look at the principles of axiomatic design provides an explanation for this, however. It would appear that two design parameters (one for each block) were each involved in the two mapping interactions, giving rise to a coupled design (in the axiomatic sense of the word as described in Chapter 2). Each mapped function ideally required a different setting of the same design parameters. In effect, a change in the one design parameter implies a change in the other design parameter (located at the other block) and this in turn affects the setting of the first one.




▸ **Figure 6.10** A closer look at the interference between interactions

Axiomatic coupling refers to a trade-off between the two parameters and in that sense is different from a single generate-test cycle. It seems very similar to our interpretation of Thompson's reciprocal dependence (see 3.1), and the interdependent interaction of the DSM studies (see 3.2) These organizational theories both argue that this interaction type is involved with intense iterative mutual adjustment.

Since building blocks were modeled as black boxes (see the discussion in Chapter 5) this taxonomy is not suitable for identifying this type of coupling. In order to do so it would be necessary to not only identify the reason for two blocks interacting, but also to precisely model the particular design parameter (location, size, material etc.) involved with that particular interaction. This would, for example, involve modeling blocks B and F with a mapping interaction, which would includes the position of component B1 and the geometry of component F1. To that end, the building blocks should be investigated at the smallest level of detail in order to explore all of the interacting detailed design parameters.

It should be noted that the interference between the interactions is not just a matter of mapping types of interaction, but can apply to all types of interactions. The likelihood that a mapping interaction will cause interference is possibly relatively high since this type generally involves detailed restrictions on single or multiple design parameters.

This investigation permitted identification of the interference between interactions and corresponding coordination needs. However, if the need arises to focus purely on the interference between the interactions, the current taxonomy will need to be extended, the building blocks opened up and all interior design parameters modeled. This may constitute a topic for further research. The drawback of including all these details is, however, that it would be at the expense of a general overview of system-level interactions, and would probably become a very technical and difficult exercise. This was precisely this researcher's objection to the axiomatic approach as stated in Chapter 2.2. This issue will be looked at again in Chapter 7.2, which deals with further research.

*Time and cost issues*

With regard to building block G, it was noted that its constraint on space was very tight. This resulted in a number of redesigns in order to achieve the goals (technical and otherwise). It was stressed that although team G had a hard task, not much additional system-level coordination was required and the team eventually reached its goal. The analysis in this research would probably have been different if time and/or cost issues had been considered. In fact, if team G needs more time to achieve its technical goals than is planned for, it will have failed to meet its objectives and additional system-level coordination is required. However, only the technical aspects of goal-setting were considered.

*Overall view and final remarks*

Overall, the taxonomy of interactions could usually be linked to specific coordination activities, which is an important finding in itself. It would seem that the technical constructs of the prescriptive literature (mainly axiomatic design literature) are appropriate for describing and analyzing the need for system-level coordination. Despite the prescriptive models paying little explicit attention to coordination aspects, this is not completely unexpected. It seems obvious than if the technical constructs had only been useful for efficient problem solving and not for easy coordination between a number of designers, the prescriptive models' claim to be based on good practice would never have been accepted.

The observation has also been made that as expected, in real life many different issues play a role where coordination is concerned. In almost all of the cases, significantly more coordination activities were performed than the prescriptive propositions would suggest are necessary. The prescriptive characteristics should hence be regarded as representing the lower limits of required coordination. The mapping interaction in particular comes to mind, since at the very least, it has a high need for mutual adjustment. The theoretical and empirical findings of the functional and global constraints show a relatively low need for mutual adjustment. For the interface and side effect interactions this was less evident, but this was expected beforehand.

These findings may be regarded as contributing the first steps of a theoretical model based on a particular architecture and able to explain or predict system-level coordination during a design process. The advantage of the constructs is currently that while these are relatively easy and objective, they are still suitable for identifying the need for coordination.

The aspects that have been suggested for further development of the taxonomy will be looked at again in the next chapter, where options for further research will be formulated. After that, those methodological aspects that also apply to the conclusions in the present section will be dealt with.

### *6.6.3 The generation of options for improvement: discussion*

Perhaps most interesting aspect of this research is that it has resulted in a real practical understanding and generated a significant number of improvements that are being (or will be) implemented by the company. The focus on the underlying technical structure of the design process (based on the taxonomy) is, in this researcher's view, its most unique feature. The taxonomy applications will be discussed in a similar fashion as the description of the results: general management, improvement per interaction, and contingencies. It should be noted that since all of the aspects have to be considered in relation to each other, the discussion will simply focus on classification.

#### *The focus on interaction in general: discussion*

At a general level, documentation of the interactions created an awareness of the current architecture and provided an understanding of the importance of managing these interactions correctly during the design process. Despite these being important things to have achieved, it must be questioned whether such insights can be solely attributed to the taxonomy. It is possible that a very general model of interactions would have been sufficient to draw attention to the interactions between building blocks and elicit general (and valid) suggestions for improvement such as better definition of responsibilities or more focused meetings. However, in this researcher's opinion, the strength of the interaction types as described here is that everybody was able to understand their relevance and found the shaver's described architecture convincing. A more general approach would probably have resulted in a discussion about the 'truth' of the interactions, instead of how the interaction could best be managed.

The focus was on technical interactions and an awareness of their importance was created. While having such a focus and taking measures to increase it in respect of system-level interactions was the logical step to take, it was, in fact, beyond the scope of the interaction constructs. Further research (more psychologically oriented, for example) would be welcome. It should be mentioned, however, that the findings that the design teams had a high internal focus and low external focus match the findings of Sosa and Eppinger as described in Chapter 4.

*Improvement per interaction: discussion*

The overview of the building blocks in a matrix form and the definitions of the interaction types made detailed discussion of the coordination activities of each particular interaction possible. Based on the underlying technological decisions, the project members were able to see precisely what coordination had taken place, and how it could have been done better (in their opinion). These remarks were very specific and included many aspects. Nevertheless, since specific interaction were being referred to, and it was very clear what issues were being talked about and the particular perspective that was being taken, it is this researcher's opinion that this made it possible for the discussion to stay at an objective level, without getting bogged down in subjective considerations. In short, it made effective discussion of the interactions possible.

In addition, the prescriptive logic behind the interactions (especially for the functional, mapping, and global constraint interactions) turned out to be useful for structuring the discussions and providing potential directions for improvement. The logic was helpful in suggesting precise and more effective ways of going about coordination, but also indicated when intense coordination efforts were an inevitable part of the process given the presence of a particular interaction.

Another issue worth mentioning is the general acceptance of the prescriptive logic. In general the designers of the project team did not actually use prescriptive models during the design process. In their eyes these were too restrictive and deviated too much from real-life processes of problem solving. Nevertheless the prescriptive constructs and logic turned out to be quite valuable, although the way prescriptive logic was applied in this research differs somewhat from engineering methodologies. The enormous body of knowledge was employed in a straightforward and focused manner and applied to detect the problems in existing situations. Instead of prescribing all kinds of steps and forcing designers to use a sequence of decision-making processes, the prescribed models were shown in an ideal situation and the alternatives available for improving the design process and the consequences of deviating from the circumstances made evident.

*Contingencies: discussion*

The fact that the interactions could be linked to their underlying causes and the firm's policy made the analysis much more acceptable to the company. The approach not only considers more effective solutions, it also indicates whether these are realistic within the current way of working or policy. Related to this, the management mentioned that they liked the two different options for improvement (coordination and technical manipulation). If there had only been a focus on the more modular aspects of structure of the product, this would probably have created much more opposition to the analysis since there was a general awareness of the real limitations involved in going more modular. The options for better coordination were hence very welcome since these provided an alternative that in terms of production is cost-free.

An unexpected effect was that the analysis not only links the product to the organization, but also bridges operational and strategic levels of decision-making. In fact most designers were individually aware that the amount of coordination was highly affected by decisions to minimize unit cost (for instance). These strategic decisions were made at a higher level and had to be accepted. However, the impact of all joint interactions on coordination make the characteristics of coordination on the 'floor' a matter of more strategic importance and ultimately a reason for reconsidering technical contingencies in general. This is possible since for each interaction its technical cause can be deduced. Each individual designer's knowledge has been collected and linked to overriding decisions made by top management and the marketing department. This also works in the opposite direction (from the strategic to the operational). Having discussed the role of this analysis, some methodological consideration will now be addressed.

### *6.6.4 Methodological considerations*

Like 6.1 this section explores the validity and reliability of the findings. The most interesting question here is whether it can be said that the results are entirely due the approach taken in this research, and in particular the taxonomy of interactions.[4] the unique influence of the taxonomy, the impact of the investigators, the effect of the analysis having a the retrospective nature, and the unique conditions of the case will be discussed below.

#### *The influence of the taxonomy*

A considerable amount of study of theory went into the development of the new taxonomy. In the first section the advantages for data collection were mentioned. There are also good reasons for claiming that the taxonomy was the most important factor behind the outcome. The logical question now is (as previously briefly noted) is whether the findings and improvements are uniquely the result of the taxonomy, or could also have been accomplished by any other interaction model. The only way a fully valid answer to this question could have been obtained would have been by comparing the results of different interaction approaches within exactly the same case situation and conditions. This is obviously not possible, and logical reasoning about why alternative interaction models (the DSM models and the existing taxonomies) would not have achieved the same results had to be relied upon.

This researcher strongly doubts that the existing interaction studies could have achieved similar results. As described in Chapter 4, the DSM studies lack a clear technical background in respect of interactions. It is thus argued that the mutual agreement on the findings, the options for better coordination modes, the options for manipulation of the interactions, as well as the clear link with the overall contingencies would not have been possible based on these interaction constructs. The same applies for the taxonomy that Pimmler and Eppinger devised. Since this taxonomy does not include the mapping or global constraint interaction it cannot be convincingly linked to architectural decisions. The effects on coordination could not have been recognized, and the options of manipulating the interactions and the relationship to the overall company policy would not have logically followed from this taxonomy.

This researcher has no qualms about attributing the results to the taxonomy and its logic. This is not to say, however, that if this analysis had not been performed, the company would have continued their traditional way of working. It would probably have had another look at its architecture and way of working, but in a different way.

[4] The effectiveness of the approach has very recently been illustrated at the hand of a steam iron design process. Similar results were found, which added to confidence about the functioning of the taxonomy. The interaction types corresponded to the same effects for coordination, and the representation itself generated options for improvement. These findings are, however, not included in this dissertation for reasons of confidentiality and the late start of the investigation in terms of this report's publication date. The research's findings relate solely to the shaver case study.

*The influence of the investigator*

It cannot be denied that that this researcher's role in the research may have had an impact on the results. The analysis was presented as enthusiastically as possible and the importance of the interactions was continuously stressed. This is unavoidable when the aim is effective and in-depth feedback on the analysis. Personal involvement and a good personal match with the members of the company are necessary conditions for doing this type of analysis, though only a necessary and not the only condition. This researcher would argue that the options for improvements were not influenced by the personal involvement, but were largely based on the logic of the interactions as defined in the taxonomy. In fact, the members of the organization frequently mentioned how correct the logic of the analysis was and how preferable to accepting recommendations on personal grounds. The project members were highly educated and well experienced and had a good eye for the matter. The fact that the company has started implementation of the options for improvement is perhaps the most impressive demonstration that the members of the organization were serious about the analysis and its logic. It is perhaps worth noting that the researcher does not have a technical background and based his recommendations on the logic of the taxonomy.

*The influence of the analysis having a retrospective nature*

Most of the data relating to coordination was collected retrospectively by interviewing project members. The results are based on personal interpretations of past instances of coordination spread over quite a long period of time. The results may be biased by the interviewees' personal impressions and also by the researcher's questions and focus on the interactions. It is obvious that the researcher's attendance could not have influenced the course of the actual design project since it was virtually finished when the research project started.

The research's strongest element is undoubtedly the high validity of the taxonomy of interactions. A clear distinction could be made between what needed to be coordinated (the interactions) and the coordination itself (between the teams). The interaction constructs and underlying theory were very useful in helping to identify what coordination was required, and this also improves the validity and reliability of the elaboration of coordination effort. Due to the interactions having a technical background, it was possible to check the technical reports for the 'facts' involved in the interactions, and facts were the items on the meeting agenda. Furthermore, reactions to the presentations and interviews with key informants could be linked up on the basis of technical issues. These multiple sources of evidence all played a part in the findings. There was particular attention paid to verification of the results during the presentations, since it was a public well known for its knowledge and open attitude to discussion, as mentioned above.

*Specific conditions of the case*

The results of this investigation were all based on a single case study and therefore particularly sensitive to the specific conditions of the case. Different results may well have been obtained with other cases.

The case study satisfied the preferred conditions noted in Chapter 5. The product was not entirely modular, the project team was large and it was split up into smaller design teams.

An important factor was that our investigation came at a good moment in time, when the 'pain of complexity' was being seriously felt. The fact that the company was very open and did not shy away from looking at options for improvement was beneficial. An additional and powerful factor was that the traditional way of conducting system-level coordination (within the shaver company) left much to be desired. This may unique to this company. However, based on the papers previously described, the system-level interaction problem would seem to be a very common one in development projects, especially when the new products are innovative (Henderson & Clark 1990).

A detailed description of the characteristics of the case and its background in order to provide an idea of the specific conditions was one of this research's aims. However, as already mentioned, how the taxonomy would function in another situation can only be decided at the hand of additional research.

### *6.6.5 Measuring of the effect on improvement: discussion*

This research commenced with a comment on the importance nowadays of companies having efficient and well performing design processes. This has some relevance for the investigation. The logic of engineering design models and organizational principles has demonstrated that the performance of design projects involving large physical products can be enhanced. Reducing the interactions between design teams, or better management of the system-level interactions must logically result in better performance. There is thus a good chance that the suggested measures for improvement will increase efficiency. The fact that the company proceeded with the recommendations contained in the analysis strongly indicates that there is a likelihood that the expectations will be fulfilled.

On the other hand, this expectation can only be stated, and not proven. The research's time frame was too short to measure whether the recommendations have had effect. As a first step towards a solution, a performance indicator that documents the percentage of fully specified interactions has been suggested. Its role would be twofold. For management, it would function as a way of monitoring to what extent improvement in the management of interactions has been booked. It also has the potential to relate the effects of the analysis to the overall performance of the design project. Figure 6.9 showed that it can be anticipated that the 'newer' projects will have specified their interactions at an earlier stage of the design process. When this phenomenon can be observed in practice, it will be possible to check whether this 'improved' curve correlates to more speed, quality, or lower project costs. There are, however, quite a few problems related to validation of the claim that the analysis will result in better project performance. The following will be mentioned:

- Many of the ongoing changes were outside the researcher's influence.

Since the presentation of the results, many things have changed within the company. A large number of improvement projects have been instituted in order to increase the performance of the design projects. These were concerned with knowledge management, IT solutions, multi-project project management and so on. These projects can all be expected to have a positive impact on project performance, making it difficult to distinguish them from the effects of the architectural improvements. In addition, many people, including management, will have changed their jobs, which may also have a significant impact on the performance of the projects.

- No opportunity for experiments.

As a possible solution to the problem of distinguishing the above-mentioned 'disturbing' factors from the effects of the analysis, this researcher considered doing an experimental design involving an experimental setting in which a project is influenced as much as possible by the recommendations of the analysis, and the other is not. When both of the projects had run their course the performance of both could be compared, and the differences explained in term of the effects of the analysis. However, the company preferred not to go along with it, and moreover, it was virtually impossible to implement. The company advocated distributing the analysis to all design projects. Furthermore, the exchange of knowledge between designers in the various teams was by nature too intense to be able to isolate one group from another. These same methodological difficulties have been noted in medical studies, for instance. Experimental research for new medicines for life-threatening diseases (e.g. HIV) have been hampered by the various groups of patients starting to exchange medicines since it is unacceptable that one group not be cured on methodological grounds.

To sum up, the performance indicator seems to be promising, but many methodological difficulties still have to be solved. The claim to increased performance will have to remain one which logic alone proves.

# 7 Conclusions and suggestions for further research

This research has devised an approach to analyzing system-level coordination from the point of view of product architecture. The analysis was conducted using the design process of an electric shaver. The findings confirm the belief within product development literature that system-level coordination within design processes is often not fully understood and is an important variable for improving project performance. With regard to this case study, it emerged that an explicit focus on the underlying architecture of the shaver increased the practical and theoretical understanding of system-level coordination. Moreover, the analysis resulted in a significant number of structured options for improvement. These included improved ways of achieving coordination for the given product architecture and options to adapt the product architecture so that coordination is facilitated. Some of these options have been implemented within new design processes for shavers, and it should be possible to measure their impact on project performance in the future.

The present chapter is the last one of this thesis. The first section presents the conclusions and the second suggestions for further research.

## 7.1 Conclusions

This research started with the assumption that efficient design processes are a requisite for company survival in this day and age. It is thus important for scientific researchers to focus on understanding and improving product development processes. Current research into successful design processes is generally separated into two separate fields of knowledge – engineering design communities and management communities. The first typically focuses on detailed technical issues concerning the construction of physical products, and the second considers how people can work together effectively. Further, several authors have stressed on a general level that technology and organization are closely related and that a clear match between the structure of the product and the structure of the organization is of crucial importance for good project performance. The relationship between product architecture and organization was then explored at a more applicable level. The scope of the research was then defined and the following research question formulated:

> *How can the particular architecture of a product be represented such that it offers a clear understanding of the characteristics required for system-level coordination during the design process, and such that it provides a vehicle to generate options for improving the performance of future design projects.*

To answer this question, the thesis was divided into two sections, a theoretical one and a practical one. The theoretical section investigated current knowledge in the fields of engineering design (Chapter 2) and organizational design (Chapter 3) and finally focused on research that combines the two (Chapter 4). Taking this as a basis, Chapter 5 developed a means to analyze the organization of design processes from the perspective of product architecture. This approach links the organizational and technical bodies of knowledge at a detailed level and forms the foundation of the practical part of the thesis. Chapter 6 then illustrated the approach and examined how it functioned within the real-life design process for an electric shaver.

The conclusions of this thesis will be structured in line with the outline given above. First, the theoretical conclusions that ultimately resulted in the proposed approach will be presented. This is to all intents and purposes the answer to the research question posed above. Second, the application of the approach to the case study will be considered and the questions as formulated in Chapter 4 addressed. The conclusions will be presented under the following two headings:

- Theoretical conclusions.
- The conclusions of the application.

### *7.1.1 Theoretical conclusions*

The following section describes how the research resulted in a proposal to analyze design projects. Four logical steps that can be considered as supportive conclusions of the exploration and discussion of the theory are first presented. The proposal is then set out.

The first important step in linking both perspectives was the following:

- Product architecture and design project teams can both be represented as systems with interacting elements.

This means that product architecture can be modeled as a set of interacting product building blocks, and a project team can be thought of as a collection of interacting (smaller) design teams. According to the DSM approach, these ‘systems’ can be represented in great detail in a matrix. In the case of a product, it is easy to see which building blocks make up the product, and how each pair of building blocks can interact. Different types of interactions can be modeled in order to distinguish different technical reasons for the interaction. Similarly, it can also be shown how a project team is split up into design teams and how they exchange information in order to achieve system-level coordination between the teams. These models are used in product development literature to deduce possible improvements for future design projects. They propose constructing more independent teams and establishing mechanisms to improve system-level coordination. The second step towards linking architecture and organization is:

- Effective product development projects match product architecture and organization.

Current research in the field of product development linking product and organization stresses that effective companies organize their design teams around the product building blocks. The structure of the product reflects the structure of the organization. As a result, the interactions between the building blocks can be mapped to the interactions between the design teams. This leads to the following step:

- Interactions between building blocks are the main reason why system-level coordination between teams is required.

If it is assumed that development projects organize their design teams around the building of the product, then interactions between the blocks can be considered to be the cause of system-level coordination. Put differently, by mapping the product onto the organization one can distinguish what needs to be coordinated (the interactions) from how system-level coordination is achieved (between the design teams) during the project.

This concept is the main vehicle for understanding and improving system-level coordination. First, the need for system-level coordination can be explained by the characteristics of the existence of interactions between building blocks. Second, system-level coordination can be improved by reducing the interactions, and thus decreasing the need for system-level coordination, or by proposing appropriate coordination mechanisms that fit the characteristics of the interactions. The next step is based on classic organization theory:

- Goal-setting is an efficient coordination mechanism.

Classic organization theory states that coordination can be achieved in different ways. The most preferable within uncertain environments is the strategy of goal-setting (after Galbraith). That is, two interacting teams specify goals on a high level of abstraction such that the detailed actions can be performed in isolation of each other. If both teams achieve their goals, the work is well-coordinated. Should the teams not be able to apply goal-setting, considerably more effort is needed to integrate the work of two teams. Furthermore, based on the work of Thompson, it is argued that the way that a coordination mechanism can be applied is determined by the characteristics of the interaction. This leads to the following:

- Whether the representation of the architecture is adequate for analyzing system-level coordination depends on which types of interactions are distinguished.

It emerged that in order to establish a systematic and meaningful analysis it must be possible to define the interactions such that for each (type of) interaction it is possible to identify its cause and its impact on coordination, and to identify manipulation options.

It was concluded that the available interaction models do not satisfy these criteria. The taxonomy (after Pimmler and Eppinger) that is usually used in technical literature to represent architectures cannot be clearly linked to (architectural) decisions and is unable to indicate characteristics for system-level coordination. Alternatively, the DSM models are only able to make a limited analysis and interpretation since their interaction construct (exchange of information) is much too global. The following answer to the research question thus emerged:

An appropriate way of representing a product architecture that enables the system-level coordination to be analyzed is to document the product building blocks and identify three types of interactions between them. These types are:

- *Functional interaction*: Two building blocks need to exchange energy, material, or information in a functional way.
- *Mapping interaction*: Two building blocks together fulfil the same function.
- Physical interaction, which comprises three subtypes:
  - *Global constraint interaction*: Two building blocks are both subject to the same constraints (e.g. on space).
  - *Side effects interaction*: Side effects (e.g. heat, vibration, magnetism) from one building block influence the functioning of the other building block.
  - *Physical interface interaction*: Two blocks physically exchange energy, material, or information and/or are attached to each other.

This taxonomy is based on a detailed investigation of engineering design literature. The following definitions must be used:

Functions are 'design goals', or 'what needs to be achieved by the product', without describing how it is achieved. A function may be expressed as a transformation of energy, material, or information, but this is not essential.

Building blocks are collections of the physical characteristics of a product that are required to achieve the functions.

It is further assumed that generate-test cycles are needed to find physical characteristics that appropriately satisfy a function.

The proposed types of interactions are directly based on the architectural decisions that determine product architecture (after Ulrich). As a result, the cause of each interaction and the way an interaction can be manipulated (by which decision) is completely transparent.

Furthermore, based on the characteristics of the types of interactions the following propositions were theoretically deduced:
Two teams whose blocks functionally interact are able to apply goal-setting.
They have the freedom to make detailed design decisions completely independently of each other as long as they achieve the functional specifications of energy, material, or information input.

- Two teams whose blocks have a mapping interaction are hampered when applying goal-setting.

Since both blocks are needed to fulfill a function, they cannot make detailed decisions independently of each other. Generate-test cycles have to be collaboratively performed to find a solution for the mapped function.

- Two teams whose blocks have a global constraint interaction are able to apply goal-setting.

A constraint can be decomposed into a smaller constraint for each block, and each team can make all design decisions completely independently of each other as long as each block satisfies its constraint.

- Two teams whose blocks have a side effect interaction need to solve an exception to the original specifications.

This coordination cannot be planned for and is reactive. It is not possible to make a clear statement about the level of detail at which coordination has to take place.

- Two teams whose blocks have a physical interface are limited in applying goal-setting.

The coordination has to take place at a low level of detail, but it is difficult to make a general statement about its implications.

### *7.1.2 Conclusions of the application*

The conclusions of the practical part will be structured according to the questions formulated in Chapter 4:

- Is the representation of product architecture fully understood, and can each interaction be linked to system-level coordination?
- What system-level coordination activities go with each type of interaction, and are the premises behind the coordination characteristics per interaction type valid?
- Does the analysis result in options to improve the design process, and what are these options?
- What are the effects of these options on project performance, or, at least, how can they be measured?

In order to obtain answers, the analysis has been applied to an virtually finished design process for an electric shaver. The project team was large and comprised smaller design teams that each was responsible for the design of a building block for the shaver. The whole project team was housed in one room. The shaver was very innovative (for both market and company) and was not fully modular. The project was finished within the planned schedule and was a great success in the marketplace. However, the process was exceedingly complex. In broad terms, the following steps to collect the data were performed:

- The shaver was decomposed into ten building blocks.
- The interactions between each possible pair of building blocks was documented according to the proposed types of interaction.
- The system-level coordination effort per interaction was retrospectively identified.
- All of the findings were modeled in a matrix and options suggested to improve the design process according to the logic proposed here. This was then presented to the company.
- The researchers were involved with the implementation of the improvements.

Each of the above-mentioned questions will be considered in the sections to follow. The role of the taxonomy will first be focused on, and then the case-specific findings.

*Common understanding of the interaction constructs*

Thanks to their clear technical background, the identified interactions were very easy to understand. During the documentation of the interactions of the shaver, all interviewees recognized the same interactions. In addition, the presentation of the interactions by several parties within the company showed a high level of consistency and an understanding of the interactions. The main advantage of the constructs was that they were directly based on a physical product and could be checked at any time.

Furthermore, the system-level coordination activities could be clearly related to the interactions. Apparently, the clear technical definitions of the constructs enabled the project members to project their specific coordination activities onto each interaction separately.

*Propositions approved*

Since each interaction could be linked to coordination activities, the characteristics for coordinating all the documented interactions could be compared. Each interaction was compared with the propositions and it was found that although the actual coordination effort could be higher than the propositions, it was never lower. This is as expected since it is known that real-life processes are never as optimal as the prescriptive logic. The following was concluded:

- A mapping interaction between building blocks needs significant system-level coordination between the design teams during the design process.(Goal-setting is hampered)
- The functional interaction and the global constraint interaction minimally correspond to a low need for system-level coordination between the design teams during the design process. (Goal-setting can be applied)
- Intense coordination effort was measured for the interface and side effect interactions but these differed per interaction, and no clear general statement could be suggested (in line with the propositions).

The most spectacular finding was that the mapping type interactions for each observation corresponded to intense system-level coordination.

These findings contribute to the first steps of a theoretical model that states that interaction type A implies coordination type B. By using the particular architecture of a product, it can explain the amount of system-level coordination that is required and suggest effective organizational structures. In addition, the propositions per interaction also play an important role within the generation of options for improvement, which will be described below.

*Understanding and improvement*

Perhaps the most interesting outcome of the research is that it resulted in a significant number of improvements. There were three different aspects in the analysis: it put increased focus on the management of interactions in general, it provided a means to improve the separate management of each interaction, and it created insight into the underlying contingencies of the architecture. These conclusions are addressed below.

*More focus on system-level coordination*

The analysis of the architecture of the shaver significantly emphasized the problem of system-level coordination within design projects. Previously, such awareness and its relationship to the architecture of the product was much less available within the firm. It was concluded that a detailed and clear overview of the interactions triggers structural discussion and encourages reflection on the general way of working. This resulted in a number of insights for the project team. They all agreed on the effective structure of design teams and advocated the one-room approach to achieve easy information exchange. On the other hand, although the company became aware that system-level interactions were insufficiently managed, the focus within each design team remained much higher than the focus on the interactions between the teams. One possible explanation is that the easy communication within the one room has a side effect in that it does not force designers to effectively and formally manage the interactions (where possible). In addition, the number of interactions between the blocks was much too high to be managed by a single hierarchically higher lead designer. The suggested measures for improvement were all in the direction of more focus on the interactions and better responsibilities for the interactions during the design process.

*Improvements for each interaction*

In addition to this general insight, the documentation was used to discuss each documented interaction separately. Based on the objective underlying structure of interactions, project members from different technical backgrounds were able to recognize and discuss what coordination had taken place and how it could have been improved.
It was thus concluded that:

- Documentation of the interactions triggers discussion and encourages reflection concerning the management of each interaction separately, which is very effective in itself.

Further, it can be stated that:

- The logic of the interactions is very helpful in structuring the discussions and guiding options for improvement.

In effect, the 'ideal' propositions assist in suggesting more adequate ways for coordinating a specific interaction, or for indicating when intense coordination effort is an inevitable part of the process given the existence of a particular interaction. Proposals were made for each interaction in turn on how they could be technically adjusted in order to simplify future system-level interaction. Hence:

- The characteristics of each type of interaction indicate more effective ways for coordinating existing interactions.
- The characteristics of each interaction type provide insight into how each interaction could be technically adjusted in order to facilitate system-level coordination.

Remarkably, options for improvement could be suggested for most of the interactions. The technical decisions were then considered, in particular within the context of coordination and goal-setting. Options for better coordination included earlier and more formal goal-setting, and preventing failure or changes to the goals. Measures to manipulate the interactions included a reduction in the number of interactions, the simplification of interactions, or the standardization of interactions. Both types of improvement could be suggested for each interaction. For mapping interactions, the options for manipulation seemed to be the most effective since all mapping interactions inevitably result in intense coordination effort. However, most of the options for manipulation are strongly limited by the company's overall policy, as will be addressed below.

*The contingencies behind the interactions*

The contingencies behind the interactions can clearly be deduced. Using the characteristics of the interaction types it is possible to discuss the reasons for the underlying decisions. It can thus be concluded that the typology is able to demonstrate how existing interactions are the result of the overall policy or way of working of the company. In the case of the shaver, it can be noted that most of the interactions could be explained by the policy of low unit cost, high emphasis and priority on styling, the available assembly system, and the established production structure. The ability to deduce the underlying contingencies of an interaction structure was considered strongly beneficial. First, it makes the analysis more realistic since the impact of coordination-friendly advice on other aspects of the company can be outlined. Furthermore, if the overall policy remains unchanged, it is clear which interactions will logically return in future projects. This insight could never have been achieved with traditional DSM models.

To sum up, the analysis resulted in a large number of lessons learned that can be applied to facilitate future system-level coordination and be expected to increase the project performance.

*The effects of the analysis on the firm*

The proposed advice has had an considerable impact on the new product development policy of the firm. After the presentations there was much more focus on the interactions during the project and more formal ways for coordination were introduced. Most importantly, an architecture team was instituted whose goal is to reconsider the current architecture and to propose a new improved architecture. The scope of this team is wider than the focus here since all aspects of the business have to be taken into account.

Based on the theoretical reasoning outlined here, these measures can be expected to initiate increased project performance. Due to the time limit of this research, it was not possible to measure the effects, however. A performance indicator that documents the extent to which the lessons learned are being implemented was proposed. This indicator can be linked to the performance of design projects in the future. This may ultimately validate the claim of improved project performance. It should be remembered, however, that there are still many methodological difficulties.

To sum up, the proposed taxonomy of interactions appears to be an adequate means of analyzing and suggesting options to improve system-level coordination. Its underlying logic also makes its application within other cases a viable option. However, future research has yet to prove these expectations.

Furthermore, the specific findings of the shaver case can only be considered on their own merits. It cannot be concluded that all design projects have a low focus on system-level coordination. However, the findings do add one more example to the literature demonstrating that system-level interactions are often poorly understood and provide an important variable for achieving better performance.

## 7.2 Directions for further research

After having addressed the conclusions, suggestions for further research will now be made. Before doing so however, it would be useful to take a brief look at the present and past situations, in line with the philosophy of DSM and this thesis.

When looking back it can be concluded that this research has a strong focus. We concentrated on system-level coordination and have applied our approach to only one case. During the research many issues, ideas, and observations passed the review. The concept of product architecture is extremely wide and highly relevant. Many interesting observations could be made during the exploration of the case, for example about the management of design processes, the many human aspects, the broad implications of architecture, and, obviously, the entire process of creating a new architecture.

The research resulted in a great number of interesting and stimulating aspects, and without doubt this thesis contains only a small part of everything that was been said and done. Despite a broadening of the scope, however, we deliberately confined ourselves to the theoretical foundation of product architecture. This was an important step in achieving focus and validity for the results, and was of course interesting in itself. In addition, we believe that the taxonomy of interactions and its ability to represent and interpret the architecture of a product offers a strong foundation for further research.

Broadly speaking, promising directions for further research include the external validation of the approach within other cases, fine-tuning or extending the approach, using the representation to study alternative implications for product architecture, the construction of new architectures, or applying the taxonomy in non-physical situations.

### *7.2.1 External validation*

As already indicated in the introduction to Chapter 1, a multiple-case-study setup is required to validate all the claims that are made in this research based on the analytical generalization of a single case study. There are a number of options:

- To further test and develop the tentative proposals about how the several types of interactions relate to system-level coordination.

For instance, is a mapping type always accompanied by a great need for mutual adjustment?

A multiple-case-study setup with contrasting product architectures will probably form the best basis for proving such a claim. Modular architectures can then be expected to have a low need for mutual adjustment, and integral ones an extremely high need. The ultimate goal is thus to construct a Thompson-like theory that is able to state which type of interaction fits which type of coordination. Such a theory can be used to predict coordination needs, or to find appropriate organizational structures for a particular product architecture.

- To check the functioning of the tool in its role of improving design processes within other cases.

Does it work in the same fashion? The findings can then be separated from the specific characteristics of the shaver case, as well as our specific involvement. If the analysis results in the same type of improvements, this would be strong evidence in favor of the approach. If the analysis yields strikingly different results, it would be interesting to identify the specific reasons that affect the approach. Do they depend on the skills of the investigator, on the specific way coordination was achieved within the case, or is it part of the representation itself? Furthermore, it would be useful to investigate whether the approach is appropriate for other organizational structures (i.e. functional organization), other product architectures, or even other types of products (i.e. software).

- To further validate the proposed performance indicator.

As discussed above, the performance indicator provides a vehicle to link the management of interactions to the performance of design projects. A couple of methodological remarks were noted but observations of different cases over a longer period of time should result in very interesting knowledge.

### *7.2.2 Refinement or extension of the taxonomy*

During the discussion of the results, several aspects that were observed within the case were noted but could not be directly explained by the interactions. A number of extensions that in turn could be the object of further research were suggested, including:

- Increasing the link with axiomatic design to identify coupled design parameters.

An interesting observation (discussed in the previous chapter) is the interference between interactions. The combination of several interactions may cause iteration between design parameters. It was concluded that in order to enrich the analysis, not only the reason why two blocks interact should be identified, but the exact design parameter (location, size, material etc.) involved with that particular interaction should also be modeled. For instance, blocks B and F have a mapping interaction that includes the position of component B1 and the geometry of component F1. To this end, the building blocks should be 'opened' into the lowest level of detail to explore all interacting detailed design parameters. The drawback of going into more detail is that it hampers a clear overview and becomes a very technical, complicated exercise. This was the main criticism of the axiomatic approach stated in Chapter 2.2. Hintersteiner and Friedman (1999) recently proposed an axiomatic design solution to identify coupled (interacting) design parameters between building blocks.

- Investigating further the probability that a mapping type of interaction can cause interference with other interactions.

This may be an alternative solution to the above assumptions. Since mapping interactions generally include the detailed specification of multiple design parameters, they are perhaps most likely to cause interference with other design parameters. One could decide to focus on these interactions in particular when searching for 'coupling' of building blocks.

- Including considerations of sequence.

The current types of interactions take no direct account of time or sequence. In order to capture the specific sequence of how a company works, the taxonomy of interactions presented here may be added to a traditional DSM approach. Interactions between designers (exchange of information) can be enriched by also specifying the type of interaction between the building blocks they refer to.

- Adding considerations of planning.

Within this analysis it is argued that a global constraint interaction causes no system-level coordination as long as the teams achieve the goal and satisfy the constraint. However, there is no mention of whether that would take a day or a year. Hence failure of a team may also occur when they exceed the time limit, which in turn causes additional system-level coordination. This is not modeled in the current taxonomy. Conceptually speaking the same applies to cost and quality variables.

- Enhancing the types of interaction by including probability of failure or slack.

These aspects turned out to be important in explaining the amount of system-level coordination involved in functional and global constraints (see the discussion in the previous chapter). These issues play a leading role in Galbraith's theory. The downside of adding these features is that the taxonomy becomes more complex and less objective. What is a probability of failure, and what is slack? These factors are probably strongly dependent on the personal opinions of the interviewees.

- Changing the approach from retrospective to predictive.

Using the strong theoretical foundation for the interaction types, it would be nice to find out the extent to which it is possible to predict the amount of system-level coordination that is required for specific types of interactions between building blocks. Perhaps the above remark including probability of failure will prove of additional value here. Constructing predictive models is generally seen as the Valhalla of scientific modeling, but very few models actually manage to be predictive (Smith & Morrow 1999). This approach may do better. However, we may have to accept that prediction is very difficult within complex environments as the following (anonymous) quote indicates: 'The tragedy of life is that we understand it by looking back, but have to live it in the future.'

### *7.2.3 Alternative interpretations*

In addition, the role of types of interactions can be applied to more than just coordination. Here are a few options:

- Elaborate the role of power as a cause of an interaction structure.

The description of the contingencies behind the interactions also covered the high priority of styling that caused many interactions. In fact, these interactions are the result of a trade-off between all the wishes of the various departments. Many trade-offs can be founded on logical grounds, but others may be the result of the relative power of the departments. What happens to the structure when the production manager has the most power within the organization? It would be very interesting to examine the division of (personal) power among the various decision makers within a company and to compare this to the interaction structure. This would provide a very different perspective on the architectural literature.

- Examine the social aspects of autonomy.

The role of focus on interactions has been discussed above. Several designers in the shaver project team mentioned a 'high' focus on internal interactions and a 'low' focus on external (system-level) interactions. Sosa and Eppinger also mention this aspect in their study. Based on clear interaction constructs, it would be interesting to study the social and psychological influence of the autonomy of design teams or workers. This type of research is quite common within social science but not that based on product architecture. Similar studies have been conducted in socio-technical research within production environments. This involved the construction of semi-autonomous production cells associated with assumptions of higher satisfaction on the part of the workers. This type of research does not yet seem to be available within product development literature and the proposed taxonomy of interactions would be a good starting point.

· Examine the appropriateness of ICT solutions as a coordination mechanism.

One of Galbraith's strategies is to increase the information handling capacity by instituting ICT solutions. Similarly, digital solutions are currently very popular. Some researchers have gone so far as to propose substituting direct human exchange of technical information with remote ICT. Others stress the use of information systems as the universal solution for the coordination problem (Lutters 1998). Given the findings of this thesis, particularly the mapping interactions, it would be extremely interesting to investigate for which type of interactions ICT applications would be an appropriate coordination mechanism and for which it would not. The study by Novak and Eppinger discussed above argues that companies with integral architectures can better design their products in-house in close collaboration and leaves the strong impression that not all interactions can be handled remotely.

Furthermore, issues like standardization, engineering change orders (Terwiesch & Loch 1999) outsourcing (Novak & Eppinger 1998), Knowledge management (Sanchez 1999a) can all be further explored using the proposed taxonomy.

#### *7.2.4 A guide to design new architectures*

The taxonomy of interactions introduced here is able to generate options to manipulate an existing architecture. It does not, however, contain a framework to construct new architectures that take all aspects of the business into account. Within the literature, a number of top-down frameworks have been developed to support firms in their choice of architecture for a product that will help to achieve the overall performance targets of the firm (Erixon 1998) (Ishii 1997, Ishii 1998, Blackenfelt 2000). However, these top-down approaches have to deal with a wide diversity of relevant factors, complicating a well-considered choice of architecture. An extension of this approach may be to maximally apply established knowledge of an existing architecture in order to contribute to the selection of an adapted architecture. Based on the representation of an existing architecture it is possible to devise an approach that considers the overall effects of small changes to the established architecture. In effect, the combination of an approach that takes the existing situation as its starting point combined with a broad knowledge of the models that take the 'should-be' architecture into account, would probably be helpful in guiding the painstaking process of changing established architectures in the right direction. Recent papers reveal somewhat similar approaches (Martin & Ishii 2000), and a very recent dissertation by two colleagues in Groningen has produced promising results in its application to steam irons (Burgsteden and Wobben 2001).

#### *7.2.5 Broadening the scope of the taxonomy*

It was mentioned in Chapter 3 that the interaction constructs of classical organizational theories need some update to be of more practical relevance. Therefore it is worth exploring how the interaction constructs proposed in this thesis can be applied for e.g. an hospital. In such a case, the functions and solutions of the hospital need to be defined and needs to be investigated how the structure results in specific dependence and need for coordination between organizational units. As being a researcher at the faculty of Management and Organization, broadening the scope of the taxonomy seems a very attractive idea.

# References


Albano, L. D., Connor, J. J., and Suh, N. P., "A Framework for Performance-Based Design," *Research in Engineering Design*, 5 (1993), 105-119.

Albano, L. D. and Suh, N. P., "Axiomatic Approach to Structural Design," *Research in Engineering Design*, 4 (1992), 171-183.

Alexander, C., *Notes on the Synthesis of Form*, Harvard University Press, 1964.

Andreasen, M., Hansen, T, and Mortensen, N., "The Structuring of Products and Product Programmes," *Proceedings of the 2nd WDK Workshop on Products and Product Programmes*, Delft, The Netherlands, 1996.

Baldwin, C. Y. and Clark, K. B., "Managing in an Age of Modularity," *Harvard business review*, 75,5 (1997), 84-93.

Blackenfelt, M., "Modularization by Relational Matrices - a Method for the Consideration of Strategic and Functional Aspects," *Proceedings of the 5th WDK Workshop on Product Structuring January 2000,* Edt. Riithahuhta A., Pulkkinen A., Springer-Verlag ,2001

Blackenfelt, M., *Managing Complexity by Product Modularization,* Ph.D. Thesis, Royal Institute of Technology, Stockholm, Sweden, 2001.

Blessing, L. T. M., "Comparison of Design Models Proposed in Prescriptive Literature" *Proceedings of the COST A3 / COST A4 International Research Workshop on "The role of design in the shaping of technology*, Lyon, 1996.

Bratton, D. A, "http://www.hr-info.com/hrarticles/dbratton/consult.html," (2001).

Brown, S. L. and Eisenhardt, K. M., "Product Development: past research, present findings, and future directions," *Academy of Management Review*, 20,2 (1995), 343-378.

Burgsteden, W. A. M. and Wobben, M., "Acceleration of Product Renewal of Steam Irons by Improving the Product Architecture," Master-thesis (confidential), Faculty of Management and Organization, University of Groningen, the Netherlands, 2001.

Cooper, R. G., *Winning at New products: Accelerating the Process from Idea to Launch*, Addison-Wesley, 1993.

Cutherell, D., "Product Architecture", chapter 9 in *Product Design*, Edt. Otto, K. and Wood, K., Prentice Hall, 2001.

Eisenhardt, K. M., "Building Theories from Case Study Research," *Academy of Management Journal*, 14 (1989).

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*, 6,1 (1994), 1-13.

Erens, F. J., *The Synthesis of Variety - Developing Product Families,* PhD Thesis, University of Eindhoven, the Netherlands, 1996.

Erixon, G., *Modular Function Deployment- A Method for Product Modularisation,* Ph.D. Thesis, The Royal Institute of Technology, Stockholm, Sweden, 1998.

Ettlie, J. E., "Product-process Development Integration in Manufacturing," *Management Science*, 41,7 (1995), 1224-1237.

Galbraith, J. R., *Designing Complex Organizations*, Addison-Wesley Publishing, Reading, 1973.

Galbraith, J. R., *Designing Organizations: an Executive Briefing on Strategy, Structure, and Process*, Jossey-Bass Inc., San Fransisco, 1994.

Gulati, R. K. and Eppinger, S. D., *The Coupling of Product Architecture and Organizational Structure Decisions* MIT Sloan School of Management Working Paper #3906, Cambridge, MA, 1996.

Hartley, J.F., *Qualitative Methods in Organizational Research; a Practical Guide,* Eds. Cassel C., Symon, G., Sage, 1994.

Hauser, J. R. and Clausing, D., "The House of Quality," *Harvard Business Review*, 3 (1988), 63-73.

Henderson, R. M. and Clark, K. B., "Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms," *Administrative Science Quarterly*, 35 (1990), 9-30.

Hintersteiner, J. D., "A Fractal Representation for Systems," *Proceedings of the 1999 International CIRP Design seminar*, Enschede, the Netherlands, 1999.

Hintersteiner, J. D. and Friedman, G., "System Architecture Template," *MIT White paper* ,Cambridge, MA, 1999.

Hubka, V. and Eder, E., *Theory of Technical Systems*, Springer-Verlag, 1988.

Ishii, K., "Life-Cycle Design of Products: A New Opportunity for Manufacturing Enterprises," 2 (1997), 1-40.

Ishii, K., Modularity: *A Key Concept in Product Life-cycle Engineering* in A. Molina and A. Kusiak eds., Handbook of Life-cycle Enterprise, to be published from Kluwer (1998).

Kratzer, J., Engelen, J. van, and Leenders, R., "Structuring the Communication Infrastructure of Innovation Teams for Success," *Journal of product innovation management*, Forthcoming (2001).

Krishnan, V., Eppinger, S. D., and Whitney, D. E., "A Model-Based Framework to Overlap Product Development Activities," *Management Science*, 43,4 (1997), 437-451.

Leeuw, A.C.J. de, *Organisatiesmanagement, Analyze, Ontwerp en Verandering : een Systeemvisie*, Van Gorcum, Assen, 1997.

Lutters, D.; Streppel, A.H.; Kals, H.J.J.; "Product information structure as the basis for the control of design and engineering processes*"; CIRP Journal of Manufacturing Systems*; 27, 2, (1998), 199-204.

Malmqvist, J., "A Computer-based Approach Towards Including Design History Information in Product Models and Function-Means trees," *Proceedings of DTM-95*, Boston, MA, USA, (1995).

Malmqvist, J., Axelsson, R., and Johansson, M., "A Comparative Analysis of the Theory of Inventive Problem Solving and the Systematic Approach of Pahl and Beitz," *Proceedings of the 1996 ASME Design Engineering Technical Conferences and Computers in Engineering Conference*, Irvine, California, 1996.

Martin, M. V. and Ishii, K., "Design for Variety: A methodology for Developing Product Platform Architectures," *Proceedings of the ASME Design Engineering Technical Conferences*, Baltimore, 2000.

McCord, K. R. and Eppinger, S. D., *Managing the Integration Problem in Concurrent Engineerin,g,* MIT Sloan School of Management Working paper #3594, Cambridge, MA, 1993.

Mesarovic, M. D., Macko, D., and Takahara, *Theory of Hierarchical, Multilevel Systems*, Academic Press, New York, 1970.

Meyer, H. M., Tertzakian, P., and Utterback, J. M., "Metrics for Managing Research and Development in the Context of the Product Family," *Management Science*, 43,1 (1997), 88-111.

Meyer, H. M. and Utterback, J. M., "The Product Family and the Dynamics of Core Capability," *Sloan Management Review*, Spring (1993), 29-47.

Mintzberg, H., *The Structuring of Organizations*, Prentice-Hall, Inc., London, 1979.

Nonaka, I., "A Dynamic Theory of Organizational Knowledge Creation," *Organization Science*, 5,1 (1994), 14-37.

Novak, Sharon and Eppinger, Steven D., *Sourcing by design :product architecture and the supply chain*, Sloan School of Management Working Paper, Massachusetts Institute of Technology, Cambridge, MA, 1998.

Novak, Sharon and Fine, C. H., *Fitting Teams to the Task: Product vs Operations Improvement at Saturn and Nummi*, Sloan School of Management Working paper, Massachusetts Institute of Technology, Cambridge, MA, 1996.

Oosterman, B.J., Gaalman, G.J.C., Kuijpers, F.P.J.; *Finding Structures in Product Development*, SOM Research Report, 99A06, University of Groningen, the Netherlands, 1999.

Oosterman, B.J., Gaalman, G.J.C., Kuijpers, F.P.J., "Visualizing Interdependencies in Technical Design" *Proceedings 5th WDK Workshop on Product structuring January 2000*, Edt. Riitahuhta A., Pulkkinen A., Springer Verlag, 2001.

Oosterman,B.J. Gaalman G.J.C., Kuijpers, F.P.J., "Product Architecture from a Coordination Point of View", *Proceedings EIASM 7th International Product Development Conference*, 2000.

Pahl, G. and Beitz, W., *Engineering Design: A Systematic Approach*, Springer Verlag, London, 1996.

Pimmler, T. U. and Eppinger, S. D., "Integration Analysis of Product Decompositions," *Proceedings of the ASME Design Theory and Methodology Conference*, Minneapolis, 1994.

Pine, B. Joseph, *Mass Customization the New Frontier in Business Competition*, Harvard Business School Press, Boston, MA., 1996.

Robertson, D. and Ulrich, K., "Planning for Product Platforms," *Sloan Management Review*, 39,4 (1999), 19-31.

Sanchez, R., "Modular Architectures in the Marketing Process," *Journal of Marketing*, 63, Special issue (1999), 92-111.

Sanchez, R., *Product and Process Architectures in the Management of Knowledge Resources* International Institute for Management Development Working Paper #IMD 99-6, Lausanne Switserland, 1999.

Sanderson, S. and Uzumeri, M, "Managing Product Families: The Case of the Sony Walkman," *Research Policy*, 24 (1995), 761-782.

Schoenmaker, J. "Beheersbaarheid van Productontwikkeling en de Technische Interacties", Master-thesis (Confidential), Faculty of Management and Organization, Groningen, the Netherlands, 2000.

Simon, H. A., *The Science of the Artificial*, MIT Press, Cambridge Massachusetts, 1981.

Smith, R. P. and Eppinger, S. D., "A Predictive Model of Sequential Iteration in Engineering Design," *Management Science*, 43,8 (1997), 1104-1120.

Smith, R. P. and Eppinger, S. D., "Identifying Controlling Features of engineering Design Iteration," *Management Science*, 43,3 (1997), 276-293.

Smith, R. P. and Eppinger, S. D., "Deciding between Sequential and Concurrent Tasks in Engineering Design," *Concurrent Engineering: Research and Applications*, 6,1 (1998), 15-25.

Smith, R. P. and Morrow, J. A., "Product Development Process Modeling," *Design Studies*, 20,3 (1999), 237-261.

Sosa, M. E., Eppinger, S. D., and Rowles, C. M., "Designing Modular and Integrative Systems," *Proceedings of the ASME Conference Baltimore*, MD, 2000.

Stake, R. B., "A Hierarchical Classification of the Reasons for Dividing Products into Modules-A Theoretical Analysis" Licentiate thesis, Div. of Assembly Systems, Dept. of Manufacturing Systems, Royal Institute of Technology, Stockhol, Sweden, 1999.

Staudenmayer, N, "Webs of Interdependency: Strategies for Managing Multiple Interdependencies in New Product Development," *Proceedings of the EIASM 6th International Product Development Conference*, Cambridge U.K, 1999.

Steward, D. V., "The Design Structure System: a Method for Managing the Design of Complex Systems," *IEEE Transactions on Engineering Management*, 28,3 (1981), 71-74.

Suh, N. P., *The Principles of Design* , Oxford University Press, NY, 1990.

Tate, D., *A Roadmap for Decomposition: Activities, Theories, and Tools for System Design*, PhD. Thesis Dept. of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA USA, 1999.

Tate, D. and Nordlund, M., "Synergies between American and European Approaches to Design," Proceedings of the First World Conference on Integrated Design and Process Technology, Austin, TX, 1995

Terwiesch, C and Loch, C. H., "Managing the Process of Engineering Change Orders: the Case of the Climate Control System in Automobile Development," *Journal of product innovation management*, 16,2 (1999), 160-172.

Thomke, Stefan H., "The Role of Flexibility in the Development of New Products: An Empirical Study," *Researh Policy*, 26 (1997), 105-119.

Thompson, J. D., *Organizations in Action: Social Science Bases of Administrative Theory*, McGraw Hill, New-York, 1967.

Ulrich, K., "Fundamentals of Product Modularity," *Issues in Design Manufacturing Integration ASME*, 39, (1991).

Ulrich, K., "The Role of Product Architecture in the Manufacturing Firm," *Research Policy*, 24 (1995), 419-440.

Ulrich, K. and Eppinger, S. D., *Product Design and Development*, McGRAW-HILL, Singapore, 1995.

Ulrich, K. and Eppinger, S. D., *Product Design and Development*, McGRAW-HILL, Singapore, 2000.

Ulrich, K., Sartorius, D., Pearson, S., and Jakiela, M., "Including the Value of Time in Design-for-Manufacturing Decision Making," *Management Science*, 39, 4 (1993), 429-447.

Von Hippel, E., "Task Partitioning: An Innovation Process Variable," *Research Policy*, 19 (1990), 407-418.

Victor, B. and Blackburn, R.S., "Interdependence: An Alternative Conceptualization," *Academy of Management Review*, 12, (1987), 486-498.

Wheelwright, S. and Clark, K. B., *Revolutionizing Product Development*, The Free Press, New-York, 1992.

Whitney, D. E., "Why Mechanical Design cannot be like VLSI Design," *http://web.mit.edu/ctpid/www/whitney/morepapers/design.pdf*, (1996).

Yin, R. K., *Case Study Research, Design and methods*, SAGE publishers, Thousand Oaks, California, 1994.

# Summary (in Dutch)

Het snel kunnen ontwikkelen van nieuwe (veelal complexe) producten is voor veel bedrijven tegenwoordig cruciaal. Hierop inspelend richt dit proefschrift zich op het begrijpen en verbeteren van ontwikkelprocessen van complexe fysieke producten. Hiertoe wordt technische kennis van product architectuur verbonden met bestaande inzichten over effectieve organisatie van ontwikkelprocessen. Verschillende onderzoekers hebben in hoofdlijnen aangegeven dat een goede afstemming tussen de structuur van een product (product architectuur) en de structuur van de organisatie van een ontwikkelproces van groot belang is voor een goede project performance. Op een meer gedetailleerd niveau echter worden techniek en organisatie binnen product ontwikkeling veelal los van elkaar bestudeerd, terwijl het voordeel van een koppeling van beide evident lijkt. In dit proefschrift is een manier ontwikkeld om de organisatie van een product ontwikkelproces te analyseren vanuit het perspectief van de onderliggende architectuur van een product. De kern is hier een taxonomie van verschillende types afhankelijkheden waarmee een product beschreven en geanalyseerd kan worden.

Dit inzicht maakt het mogelijk om gestructureerd aanbevelingen te doen ter verbetering van ontwikkelprocessen. Enerzijds kan dit door de organisatie beter af te stemmen op de bestaande product architectuur, en anderzijds door de product architectuur aan te passen zodat belangrijke organisatorische knelpunten worden voorkomen. Deze methode is toegepast om een ontwikkelproject van een elektrisch scheerapparaat te analyseren. Het blijkt dat het verloop van het ontwikkelproject verklaard kan worden op basis van de architectuur van het scheerapparaat. Uit deze zienswijze kwam een groot aantal opties ter verbetering voort, waarvan enkele ook daadwerkelijk zijn (of worden) toegepast in nieuwe ontwikkelprojecten.

Dit proefschrift legt de sterk de nadruk op de theoretische onderbouwing van de voorgestelde methode en beschouwt de case-study als illustratie en exploratie van de aanpak. We zullen eerst de kern van de methode aangeven en daarna de toepassing beschrijven.

Bij het ontwikkelen van een nieuw (fysiek) product is normaliter een groot aantal mensen betrokken die ieder een deel van het werk op zich nemen. Coördinatie is hierdoor van enorm belang om het gehele project tot een goed einde te brengen. Het is de uitdaging om projecten zodanig te organiseren dat de coördinatie zo effectief mogelijk verloopt. Zeker voor grote projecten is dit niet vanzelfsprekend. Naarmate het aantal betrokken ontwikkelaars toeneemt kan de totale coördinatie behoefte tot grote hoogte stijgen. Om de totale coördinatie binnen grote projecten in de hand te houden is het verstandig om het projectteam op te splitsen in kleinere design teams. In het ideale (hypothetische) geval zijn deze designteams geheel onafhankelijk zodat ze taken volledig parallel kunnen uitvoeren. Normaal gesproken echter zijn de design teams in een bepaalde mate van elkaar afhankelijk en is coördinatie tussen de design teams een vereiste om het gehele project goed te laten verlopen. De afstemming van interacties tussen design teams noemen we 'system-level' coördinatie en heeft de speciale aandacht in deze studie. Uit literatuur blijkt dat deze interacties vaak slecht begrepen en gecoördineerd worden, terwijl dit essentieel is voor een goede project performance. Het beter begrijpen en beter afstemmen van interacties tussen design teams is daarom een veelbelovende insteek om verbeteringen te behalen. Een kleine groep recente studies werkt dit idee verder uit. Deze modelleren alle interacties tussen de design teams van een bestaand projectteam op een overzichtelijke wijze en leggen hiermee het vertrekpunt om een project beter in te richten. Gebaseerd op een bestaand overzicht van alle interacties zijn er ruwweg twee richtingen voor verbetering:

- Het toepassen van geschikte coördinatie mechanismen om de interacties tussen de design teams effectief af te stemmen.
- Het verminderen van de afhankelijkheden tussen de design teams waardoor system-level coördinatie eenvoudiger wordt en meer parallel (en dus sneller) gewerkt kan worden.

Deze ogenschijnlijk eenvoudige principes zijn attractief om te gebruiken binnen ontwikkelprojecten. De tekortkoming is echter de manier waarop interacties tussen design teams gemodelleerd worden. Bestaande studies vatten interacties op als uitwisseling van informatie tussen de design teams. We constateren echter dat deze definitie veel te algemeen is en voor velerlei uitleg vatbaar. Doordat de achterliggende reden van een interactie niet kan worden afgeleid, is de waarde van de analyse en adviezen zeer beperkt.

Dit proefschrift levert juist hier een bijdrage door het concept product architectuur toe te voegen. Product architectuur wordt gedefinieerd als de wijze waarop fysieke 'building blocks' (bouwstenen) van een product met elkaar samenhangen om het gehele product goed te laten functioneren.

Op basis van bestaande organisatorische inzichten stellen we dat effectieve ontwikkelprojecten zich zo organiseren dat de design teams zich elk afzonderlijk richten op het ontwikkelen van een building block van het product. Met dit gegeven leiden we af dat de interacties tussen de building blocks van een product noodzakelijkerwijs behoefte aan system-level coördinatie tussen de ontwikkelteams veroorzaken. Het begrijpen van (duidelijk waarneembare) interacties tussen building blocks van een product biedt zo de mogelijkheid om system-level coördinatie gedurende een ontwerpproces te kunnen analyseren. Hiermee is het verband tussen product architectuur en organisatie van ontwikkelprojecten gelegd.

We stellen ons vervolgens de vraag op welke wijze interacties tussen building blocks moeten worden gedefinieerd om zo goed mogelijk de system-level coördinatie te kunnen begrijpen. We beargumenteren dat de interacties zo gedefinieerd moeten worden dat 1) de achterliggende technische oorzaak eenduidig is, 2) de noodzakelijke consequenties van een interacties voor system-level coördinatie kan worden afgeleid, en 3) duidelijk is hoe een interactie technisch kan worden aangepast.

Met deze eigenschappen kunnen we interacties van een bestaand product documenteren, zodat we de system-level coördinatie tijdens een ontwerpproces kunnen analyseren en gestructureerd opties ter verbetering kunnen aandragen. Deze opties zijn:

- Effectievere coördinatie van bestaande interacties tussen building blocks.
- Technische aanpassing van de interacties zodat de behoefte aan system-level coördinatie afneemt.

Na bestudering en discussie van technische literatuur introduceren we een taxonomie van drie verschillende types interacties tussen de building blocks van een product. Het betreft de functionele interactie, de mapping interactie, en de fysieke interactie.

De kern van deze taxonomie is gebaseerd op het onderscheid tussen functies van een product en zijn fysieke oplossingen. Een functie van een product vatten we op als een ontwerpdoelstelling die beschrijft wat een product moet doen, zonder te specificeren op welke manier dit gebeurt. Functies kunnen beschreven worden als transformaties van input en output van energie, materiaal of informatie, maar dit is niet noodzakelijk. De fysieke oplossingen specificeren de fysieke eigenschappen van het product om de functies te kunnen realiseren. Een building block vormt op deze wijze een fysiek deel van het product en vervult een deel van de functies van het gehele product. De drie types interacties worden als volgt gedefinieerd:

- Een functionele interactie is aanwezig als de building blocks ieder hun eigen functies vervullen, maar deze wel energie, materiaal, of informatie moeten uitwisselen.
- Een mapping interactie betreft twee building blocks die samen een functie vervullen
- Een fysieke interactie bestaat uit drie sub-types. Twee building blocks kunnen betrekking hebben op een zelfde 'global constraint' (bijvoorbeeld een randvoorwaarde op ruimte), kunnen een fysieke interface hebben, of er kunnen neveneffecten (trilling, warmte, straling, etc.) optreden.

Deze verschillende types interacties tussen building blocks hebben ieder hun specifieke kenmerken voor system-level coördinatie gedurende het ontwerpproces. We betogen dat 'goal setting' een effectief coördinatie mechanisme is om het werk van de verschillen design teams af te stemmen. Hierbij wordt voor elk design team een doel bepaald die ze moeten realiseren, maar elk team de vrijheid laat hoe dit bewerkstelligd wordt. Goal-setting is echter niet mogelijk voor elk type interactie. We nemen aan dat elk design team een building block moet ontwerpen en we leiden de volgende proposities af.

- Een functionele interactie maakt 'goal setting' mogelijk. Design teams kunnen ieder onafhankelijk van elkaar ontwerpbeslissingen maken, zolang ze er voor zorgen dat hun block de juiste input en output realiseert.
- Een mapping interactie is moeilijk te verenigen met het stellen van een afzonderlijke goal voor elk team. De teams zullen hun ontwerpbeslissingen heel frequent en op een gedetailleerd niveau moeten afstemmen om de gemeenschappelijk functie van het block te realiseren. Dit zal gepaard gaan met veel iteratie.
- De global constraint interactie biedt mogelijkheid voor het stellen van een doel voor ieder team. Teams kunnen gedetailleerd beslissingen onafhankelijk van elkaar maken zolang ze binnen hun randvoorwaarde blijven.
- De coördinatie van neven effecten kan niet worden gepland en is reactief, maar het is moeilijk een precies statement te maken voor het niveau van detail waarop dit plaats vindt.
- De coördinatie van fysieke interfaces laat 'goal setting' slechts beperkt toe. Coördinatie moet op een laag abstractie niveau plaats vinden, maar het is lastig een algemene propositie over de precieze gevolgen te maken.

Verder laten we zien dat voor elk type interactie duidelijk is welke technische oorzaak erachter zit, en hoe deze eventueel in de toekomst kan worden gemanipuleerd en in welke bredere context dit speelt. In het kort illustreren we de bijdrage ten opzichte van de bestaande studies en geven we de omgeving aan waarbinnen de taxonomie het meest tot zijn recht komt.

De taxonomie wordt toegepast binnen een enkele case-study. Voor het ontwerpproces van een elektrisch scheerapparaat wordt nagegaan in hoeverre de proposities geldig blijken, en tot welke inzichten de analyse leidt.

Door middel van interviews worden alle interacties tussen de blocks van het scheerapparaat gedocumenteerd en wordt de coördinatie die met elke interactie gepaard ging (retrospectief) achterhaald. Uit het onderzoek bleek dat de interacties goed en eenduidig werden begrepen, en direct konden worden gekoppeld aan system-level coördinatie. De verschillende karakteristieken van de types interacties kwamen goed overeen met de gevonden coördinatie activiteiten. Met name de mapping interacties was zeer onderscheidend. Elke gevonden mapping interactie tussen twee building blocks kwam overeen met intensieve en langdurige coördinatie tussen de betrokken teams.

Op basis hiervan zijn opties ter verbetering afgeleid en gepresenteerd voor het bedrijf. De resultaten van de analyse hebben we op drie onderwerpen ingedeeld: algemene bewustwording van de interacties, verbetering per individuele interactie, en inzicht in de achterliggende (strategische) redenen van de interacties.

Ten eerste leverde de analyse een algemene bewustwording en discussie op van het belang van de interacties tussen de building blocks. Op basis van de duidelijk herkenbare onderliggende structuur van het product was het mogelijk de huidige project organisatie en manier van werken tegen het licht te houden. Gezien het grote aantal interacties tussen de blocks kwam naar voren dat in de toekomst meer nadruk moet liggen op een gestructureerde afstemming tussen de design teams.

Ten tweede kon elke individuele interactie gedetailleerd worden bediscussieerd en opties ter verbetering worden geopperd. De kenmerken van elk type interactie speelden hier een belangrijke rol. Voor elk type is aangeven hoe een interactie effectiever afgestemd had kunnen worden. door bijvoorbeeld 'goal setting' bewuster toe te passen. Daarnaast is geïllustreerd op welke punten ingewikkelde coördinatie onvermijdelijk is, gegeven het karakter van een interactie. Verder is beargumenteerd hoe bepaalde interacties technisch konden worden aangepast om op cruciale punten coördinatie te vereenvoudigen.

Ten derde was het mogelijk ook de diepere achterliggende redenen van de (vele) waargenomen interacties te verklaren. Hierdoor werd duidelijk op welke wijze de huidige architectuur ingebed is binnen de huidige (strategische) keuzes van het bedrijf, en welke brede afweging gemaakt moet worden om in de toekomst de architectuur aan te passen en de coördinatie te vereenvoudigen.

Na presentatie van de bevindingen, is een aantal voorstellen ter verbetering ook daadwerkelijk door het bedrijf in gang gezet met als doel toekomstige ontwikkelprocessen te verbeteren.

De duur van dit onderzoek was te kort om aan te geven in hoeverre dit daadwerkelijk resulteert in hogere performance van deze projecten. Er is een 'performance indicator' geïntroduceerd waarmee de verwachte positieve effect relatie in de toekomst gevalideerd kan worden.

De voorgestelde drie types interactie en hun eigenschappen worden als een belangrijk resultaat van dit onderzoek beschouwd. De verwachte karakteristieken zijn voor elk type interactie gestaafd in de case. Hiermee is een eerste stap gezet richting een theorie waarbij op basis van herkenning van interacties binnen een fysiek een product de gevolgen voor system-level coördinatie tijdens het ontwerpproces kan worden verklaard. Dit is een bijdrage ten opzichte van technische en organisatorische theorieen.

Daarnaast wordt geconcludeerd dat het overzichtelijk weergeven van de onderliggende architectuur van een product binnen de bestudeerde case heeft aangezet tot verbeteringen van toekomstige ontwerpprocessen. Het representeren van een productarchitectuur levert praktisch inzicht op in de behoefte aan coördinatie. De interacties creëren bewustzijn in het algemeen, kunnen op basis van logica individueel worden verbeterd, en kunnen worden gerelateerd aan de achterliggende (strategische) keuzes van een bedrijf. Bestaande studies op dit terrein geen dit inzicht niet, of in beperkte mate.

Omdat de toegepaste methode op theorie is gebaseerd verwachten we dat de resultaten ook bij andere projecten zullen gelden. De specifieke bevindingen van de case moeten op zichzelf worden beschouwd en kunnen worden toegevoegd aan de bestaande studies op dit terrein.

Tenslotte wordt voorgesteld om in de toekomst verder te onderzoeken hoe de methode extern kan worden gevalideerd, hoe de methode kan worden verfijnd, hoe de methode op andere terreinen zijn waarde kan hebben, en hoe de representatie van architectuur kan worden gebruikt om een nieuwe product architectuur te ontwerpen, en als laatste in hoeverre de taxonomie algemeen toepasbaar is om een organisatie (zoals bijvoorbeeld een ziekenhuis) te kunnen analyseren.