Improving product development projects by matching product architecture and organization

Oosterman, Bas Jeroen

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2001

Link to publication in University of Groningen/UMCG research database

Citation for published version (APA):

Copyright
Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

Take-down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): http://www.rug.nl/research/portal. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.
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).

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.

Figure 4.1 Reflecting product architecture within the organization
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](image)

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.

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.