Cristian Nilsson

A Framework for Continuous Design Reuse Management Supported by an Option-Based Reuse Approach

Abstract

This PhD thesis aims to address the problem of *missed design reuse opportunities across consecutive products* in industry. The research specifically targets the perceived knowledge gap regarding conceptual tools for reuse-related decision-making under different levels of uncertainty about the future. The research progressed according to the following method: first, we performed qualitative interviews in industry to identify and delimit the above chosen area of improvement potential. Second, we did an extensive *analysis* of literature, studying the dynamics of design reuse from six viewpoints: the process –, market –, costs –, artefact –, knowledge –, and organisational views. Also, two of the most important product development strategy trends (platform-based and lean product development) were investigated from the perspective of design reuse. Third, we did *synthesis* of conceptual tools, using abductive reasoning and combining elements from different knowledge fields, to provide our main contributions as presented below. Fourth, we evaluated the results by discussing their internal consistency and relevance to the research community and to industry.

The first of the main results is the *continuous reuse management framework*, which highlights how reuse-related decisions at micro-level (individual design solutions) translate to a macro-level flow of design solutions across product generations, driving the evolution of the product portfolio. The continuous reuse management framework is intended as a conceptual decision-support tool encouraging pragmatic, proactive, and uncertainty-aware handling of reuse. We classify the micro-level reuse approaches into three types, according to 1) whether preparation for reuse takes place and 2) whether future reuse is predetermined. These types are: ad-hoc reuse (no preparation for reuse), option-based reuse (preparation for future *option* to reuse) and predetermined reuse. The *option-based reuse approach* is the second main result of this PhD study. It can be described as a novel formalisation of the way companies reason when they invest in the reusability of a solution to provide future projects the *option* to reuse it. The option-based reuse approach borrows real-options thinking from financial theory to value future reusability of design solutions.

Acknowledgements

This Ph.D. thesis is the result of my research at the Department of Engineering Design and Materials at NTNU in Trondheim from February 2004 to March 2007. The choice of theme for the research responds partly to a drive within the department and academia in general to cover ‘new ground’ in the area of product platforms and related issues, and partly to a genuine interest in the conceptual and strategic aspects of product development. This personal interest had been awakened during my four years as product developer in the software industry, and during the research I often found myself motivated by my experiences and observations from that time. Therefore, although the results from this thesis are theoretical by nature, I have arrived to them driven by the reflections around my industrial experience and informal discussions with engineers in different industries, which I trust has kept the reasoning ‘down to earth’.

In particular, I would like to express my gratitude to my supervisor Professor Ole Ivar Sivertsen for all time and engagement dedicated to give me support, discussions and inspiration, and for always being available. I would also like to thank my co-supervisor Hans Petter Hildre for invaluable feedback and insight in the research field, Jens Røyrvik for mentorship, academic stimulation and proof-reading, and Tormod Jensen and other colleagues and friends for interesting discussions and encouragement.

I would also like to thank Rasmus Pedersen and Anne for receiving me in Copenhagen during my visits to Denmark’s Technical University and prof. Mogens Myrup Andreasen and the rest of the Ph.D. student group at MEK, DTU, for making those visits so fruitful.

Finally, I would like to give infinite thanks to my beloved Klara for all patience and for being my continuous source of inspiration.

Contents

1 Introduction

In this first chapter, we introduce the reader to the topic of this PhD thesis. We describe some aspects of the situation in industry that are of our interest, and formulate the challenge in industry that motivates this research. Finally, the outline of the thesis is presented.

1.1 The long-term performance of product portfolios

For companies to be successful in the long run, i.e. over many product generations, the aggregate success rate of the entire product history must be positive, so that revenues from earlier launched products can be used to create new products. Therefore, it is not sufficient to produce single successful products, "one-time-hits" [Drejer & Gudmunsson 2002]. The factors that determine the long-term competitiveness of product portfolios can be classified into outward-oriented, i.e. about how well the company relates to its customers and other actors, and inward-oriented, i.e. about how efficiently the company can develop and produce its products. Sometimes, some of these factors may be in conflict, for example when customers demand product features that are expensive for the company to supply. Each company must therefore find ways to offer a competitive product variety to the market while limiting the costs internally driven by such variety [Desai et al 2001].

In order for companies to achieve this, it is usually insufficient to base decisions exclusively on a 'snapshot' view of the present. Rather, present data should be complemented by an understanding of the *history* and the *projected future* of the product portfolio. This means viewing the *evolution* of the product portfolio, as a flow of products [Cooper 2001]. Such a view should help understand the dynamics of each particular enterprise and the long-term consequences of decisions. We are thus implying that what seems reasonable considering one-product-at-a-time, may be inappropriate for if a longer time span is considered.

1.1.1 The problem of variety and complexity in product development

Internal variety and complexity costs increase with the number and diversity of items and activities (e.g. components and process steps) needed by the company to produce its product portfolio [Franke et al 2002]. Unfortunately, these are often difficult to measure, trace and control. According to Andreasen et al [2001], the challenges in product development stem from two sources, one representing needs and opportunities, and the other representing long-term consequences of decisions:

1. *"The market's dynamic and gradually more detailed and specific demand for products which are fitted or customised to the buyer or user based upon more and more aspects, features, performances or dimensions added to the physical products: delivery, knowledge, services, maintenance, returning by end of life, recycling, etc.*
2. *The company's gradually growing portfolio of products offered to the market as a result of past and present product development and manufacturing activities, existing often in a vast number, with many variants, combined into many different products, some prepared for a healthy business, others leading to unrecorded losses when sold."*

[Andreasen et al 2001 p.13]

Many companies operate in markets that demand frequent new products to suit changing individual needs and make use of technology improvements. In highly competitive markets, customers demand products that are optimised for their individual needs without compromises and within a short time window. This has lead many companies to focus most of their attention on their front-of-the-line products, to meet customer demands as close and fast as possible, in pursuit of short-term revenue and market share. In extreme cases, companies risk entering vicious cycles of fire-fighting, where rapid 'fixes' are made with no consideration for future consequences whatsoever [Bohn 2000]. The risk of short-term focus is that each new product may inadvertently add to a 'luggage' of internal variety and complexity costs. Meyer claims that:

> *"Seeking to build the perfect product for each new customer group, engineers lead the corporation away from commonality. Each time a new customer request is formalized new parts are added to achieve the optimum solution without consideration of the downstream costs of the decision. The engineer, or the engineering manager, rarely gets wind of those costs. As the components of the firm's products proliferate – be they motors, fasteners, or whatever – opportunities to achieve economies in procurement diminish."* [Meyer & Lehnerd 1997 p.56]

1.1.2 Exploiting synergy effects between products

How should the above mentioned challenge be confronted? How can companies both become faster and more accurate at launching new products and at the same time reduce the aggregate costs due to internal variety? The general answer is: by exploiting *synergy effects* between the products. This means that the investments put into one product should benefit also other products. In manufacturing, this is achieved through *economies of scale*, present when a large volume of units share the fixed manufacturing costs. Synergy effects are in one way or another exploited by most successful mature companies: they do not start from scratch with each new product but instead reuse a degree of the assets developed for past products. Such reused assets can be 'hard', such as parts and manufacturing tools, and 'soft', such as working methods, personnel skills and organisation forms.

1.2 Problem formulation

What is specifically the challenge or 'problem' that we are interested in? This section provides a brief discussion thereof and a research problem formulation.

1.2.1 Design reuse

One way of exploiting synergies is to reuse design solutions between products. In this study, the term *design solution* is refers to those solutions that have been chosen in response to a design problem in the development of a product. Upon design reuse, a solution designed for a previous product is reused in a later product, thus avoiding the cost of developing and handling a new solution.

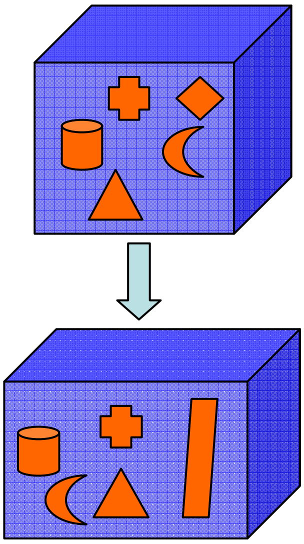


Figure 1: Design solutions carried over across product generations

Design reuse is a product-based strategy aiming to exploit as much value as possible from fewer solutions. This can be compared to process-based strategies, which aim to improve the capacity to generate and handle more design solutions:

> *"Process-based strategies seek to imbue production and distribution processes with sufficient flexibility to enable them to accommodate a high level of variety at reasonable cost. Product-based strategies seek product designs that allow high variety in the marketplace while presenting the production and distribution system with a relatively low level of component variety and assembly complexity."* [Fisher et al 1999 p.297]

Benefits of reuse has been observed in various pieces of empirical research, for example one case study of two reuse programs at Hewlett-Packard [Lim 1994], that showed improvements regarding product quality, productivity, time-to-market and costs. Defect rates were reduced due to components being proven and corrected by use in several products, and productivity rose because of less work needed, although Lim argues this does not necessarily translates to reduced time-to-market. Overall, the findings of the case studies were highly encouraging for reuse.

Companies decide which design solutions to reuse and when to introduce new design solutions. Unfortunately, the full effects of design reuse usually show in the longer term and are difficult to quantify and foresee. Hence, it can be problematic to make design reuse choices that are strategically sensible for the company, especially if there are incentives that reward immediate and tangible results. So it is apparent that an increased understanding of the effects of design reuse is needed.

1.2.2 A vision for design reuse

One long-term goal of product development could be formulated as:

Achieving a sustainable evolution of the product portfolio, so that the products are improved fast enough without the internal costs increasing too much.

Design reuse is an *instrument* that can be used to achieve this goal. To this end, design reuse should:

- make the total assortment of design solutions more coherent and transparent

- reduce ‘waste’ in product development (costs that do not lead to customer value)

So how can we envision ‘ideal reuse’ behaviour to serve as the ultimate goal for reuse strategies? Let us consider a fictive series of products that have been sequentially developed. We eliminate uncertainty by viewing these *in retrospect* supposing all the facts are known (especially development and production costs), and we assume the functional performance is given and that development lead time can be translated into costs. Then, theoretically we could find out which set of solutions that should be shared between products in order to minimise the total costs. If the company had chosen this optimal combination of common and product-unique design solutions, it would have practiced ‘ideal reuse’. However, for our vision to be more usable, we need to be more specific about how the costs and capabilities are accounted for. There are several possible simplifications we could make, of which two are interesting to consider and compare:

- One is to suppose that *identical technological knowledge* is used during the development of each of the products in the series, and that there is *no cost for transferring* solutions from one product to another. (This is roughly the situation when the products are developed simultaneously in one project.) We call this vision ‘ideal *sharing*’. We perceive that a lot of literature on design reuse implies this kind of vision, but we find it unnecessarily restrictive for the aims of this thesis.
- Another simplification is to take into account that the company possesses *different technological knowledge* when developing the different products (e.g. the company learns from previous products and from the market and competitors) and to assume the *transaction cost* of reuse is significant. Then our ‘ideal reuse’ would be the reuse that minimises the total development and production costs of the products.

In this thesis, it is the second simplification, ‘ideal reuse’, that is most relevant. This is because when considering design reuse *over time*:

- Environments and capabilities drift from the time of development of one product to the next. The technological knowledge evolves because the company learns from research, experiences from previous products, the market and competitors. Furthermore, customer needs and preferences change.
- The cost of transferring a solution from one project/product to another becomes significantly higher because of the time lag. There is no real-time feedback loop between the designers and the re-users of a solution. This transfer effort can sometimes even be larger than the effort to design the solution from scratch.

As is covered in subsequent chapters of this thesis, good design reuse practices rely on the creation of reuse *potential* and its skilful exploitation. Having a flow-oriented view of the evolution of design solutions that make up a product portfolio can make it easier to value and exploit the reuse potential.

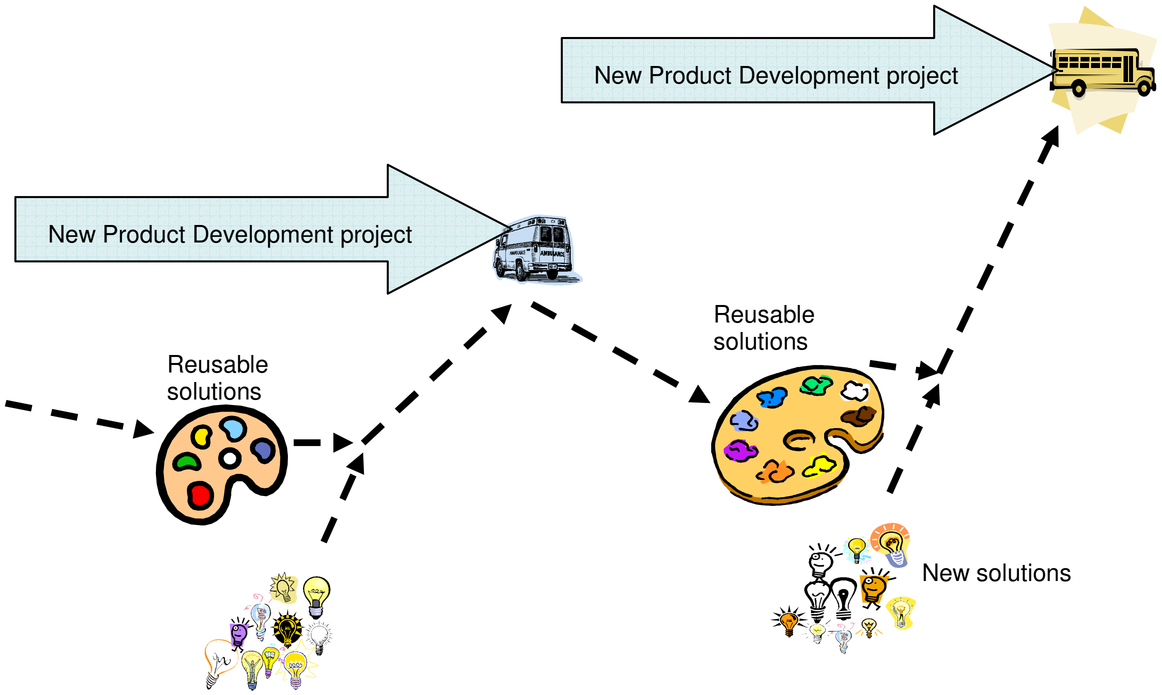


Figure 2: Each new product is a combination of new design solutions and solutions from previous products

1.2.3 Practical goal – Perceived problems in industry

The perceived improvement potential that this study ultimately intends to contribute to is the problem of *missed reuse opportunities over consecutive products*, or in another formulation, *unnecessary generational variety.* Missed design reuse opportunities can often be identified *in retrospect*, through post-mortem analysis of series of finished product development projects. This problem is significant at many companies because missed reuse opportunities lead to unnecessary costs of variety and unnecessary designing effort (increased lead time).

The problem of missed reuse opportunities is of course more relevant when the *default behaviour* is to design solutions from scratch for each new product (radically changed products), than when the default behaviour is to reuse (incrementally changed products). In the first situation, the important question is "what should we reuse?", whereas in the second, the question is "what should we change?"

The amount of potential savings through design reuse is considerably company-dependent, and for our purposes pointless to try to generally quantify. Nonetheless, it should be no doubt that the potentiality of design reuse is significant. One indication of this can be found in a study of design reuse benefits by Duffy and Ferns [1999]. Their study was based on a questionnaire to practising product developers and found that: The *current* benefits of design reuse were perceived to be 10-11% in the areas of 'reduced time', 'improved quality' and 'performance', and 6% in 'cost reduction'. Interestingly, the *potential* benefits where foreseen to be considerably higher: 25-28% for 'cost', 'quality' and 'time', and 20% for 'performance', implying a *twofold to fourfold improvement potential*, especially in the 'cost' category [Duffy & Ferns 1999].

Anticipating the analysis chapters in this thesis, we can say that one common cause for missed reuse opportunities is a lack of proactive approach towards reuse. Andreasen et al argue that

> *"Product development is missing a proper design preparation phase, where all aspects of stream building the project are treated: Focus for quality efforts, focus for re-use and innovation, choice of product and product family architecture etc. [Design preparation] is important for integrating the product quality or ambition aspect, the internal cost aspect and the business aspect. But we see only sparse signs of the proper use of this thinking in industry."* [Andreasen et al 2001 p.51]

Problems also stem from design decisions that do not consider the costs of internal variety. This has been described by several researchers. For example, Andreasen et al note that

> *"Product variety, complex activities and weak communication between development, production and distribution are eating resources. The lack of transparency and lack of cost information related to product variety and the company internal activities leads to false decisions."* [Andreasen et al 2001 p.68].

Meyer and Lehnerd provide us with a quite similar description:

> *"Diversification is often unplanned; products and product lines are added one product at a time and without the benefit of overarching strategic principles. Typically, the additional products create greater complexity in manufacturing, procurement, and distribution, and the attention of senior managers is consumed in managing that complexity. Once senior managers lose touch with the development of new products, new product strategy is guided by the initiatives of middle managers. The notion of a centrally rationalized product strategy guided by the top executives gets lost in the bureaucracy."* [Meyer & Lehnerd 1997 p.56]

This thesis aspires to give the reader increased understanding of the possibilities and mechanisms of design reuse, thus helping her make sounder decisions even under uncertainty.

Lastly, we note that despite companies usually understand the importance of variety management, there is a general lack of tools to quantify and forecast the effects of design decisions on variety [Thorntorn et al 2000]. Although this thesis does not provide any quantitative tools, it is hoped that the conclusions here presented can be used to develop such. In an interview-based study of five Norwegian manufacturing companies, it was found that design reuse indeed is generally considered of vital importance for the company [Nilsson 2006a]. Considerable efforts are devoted to improve the reusability of designs and reuse design solutions from the past. However, among the interviewed engineers, many perceived that the justification for reuse and preparation for reuse was based on 'feel' and common sense gained over long periods of being employed by the company, and not explicit rules or decision support tools. One typical statement from and interviewed engineer was

> *"The new mechanism that has been designed has cost us much, so we will try to reuse it in many products. The mechanism was made of minimal size to increase its reusability potential, through more flexible interface, but we haven't made analyses of reusability."* [Nilsson 2006a p.7]

The consequence of this is that, whenever inexperienced teams or teams with poor knowledge of the history and inner workings of the company are employed (such as consultants and sub-suppliers) there is a risk that design reuse is neglected if not explicitly required. One interviewed engineer states that

> *"One problem is that [collaboration with the sub-suppliers] often ends up with too much local focus and optimisation. We do not get the overview to find reuse at a system level."* [Nilsson 2006a p.6]

We see the need to develop terms and concepts that can be used to make reuse-related considerations more explicit, which is one of the research motivations of this thesis.

1.2.4 Scientific goal - Knowledge gap in academia

This thesis is targeted at a perceived knowledge gap in the area of *managing* design reuse in contexts ridden with *uncertainties*. (We use the term 'managing' in the sense of planning or selecting between technical alternatives, not in the sense of leading personnel.)

Despite considerable research into "product family" development in the latest decades, there is still much to be done to provide theoretical tools that help companies improve their product portfolio incrementally in a imperfect, uncertain and sometimes chaotic reality. Such research should improve the understanding of how singular design choices affect the totality of the product portfolio over time, what the difficulties are, and specifically which *mechanisms* enable the bottom-up emergence of sustainable, reusable solutions that make future variants easier to develop successfully. This need is for example highlighted by Fisher, who claims that

> *"the challenge of component sharing is increased as the decision is viewed dynamically. In most industrial situations, there already exists a portfolio of products and the managerial problem is to decide which components to re-use, which components to replace, and which new components to develop. This problem is complex and deserves further research attention."* [Fisher et al 1999, p.312]

The role of academic research is often to provide *conceptual* tools and models that the industry can utilise to develop practical tools and implementations. Accordingly, here we deal mostly with the conceptual aspects of reuse decision-making. However, the practical aspects are thoroughly analysed because they constitute a great share of the obstacles to ideal reuse practices.

By practical problems, we mean problems that companies normally know *how to* solve (known goal and method, at least in theory), but are troublesome for some reason such as being resource-demanding. Typical reuse-related practical problems include:

- gathering data to quantify the costs related to different design choices
- optimising one design solution to fit in several products with known performance requirements
- making sound design reuse decisions based on available information (under pressure)

In this study we aim especially at the conceptual problems. These are by definition abstract and thus often applicable to different cases. In conceptual problems, the goal and/or the method is unknown, i.e. there is no optimisation criteria clearly telling right from wrong. Typical conceptual problems include:

- how to know if to reuse a past solution if the consequences (e.g. costs, risks) are unknown?
- how to design for reuse when reuse requirements are unknown?

- how to justify investments for future reuse in evolving environments?

Conceptual problems are about how to deal with what is unknown. Of course, what in the beginning may be a conceptual problem to a team (“we don’t know what to do with this!”) may, when the team learns more about the matter, transform into a practical problem (“now we know what to do, but it means a lot of job!”). Conceptual problems are sometimes dealt with by means of heuristics stemming from more or less fortunate historical reasons (“we don’t know why, but we usually design like this!”).

1.3 Outline of the thesis

The thesis is structured as follows. In chapter 2 the research questions are formulated and the scientific research approach is described. In chapter 3, the phenomenon of design reuse is analysed from six different viewpoints: process, market, costs, artefact, knowledge and organisational points of view. The state of the art is presented for each of these viewpoints, and problems and knowledge gaps are identified. The goal is to provide the reader with an understanding of the mechanisms that concern design reuse at companies, both at the low-level and at high-level, in single cases and viewing the aggregate evolution of products at the macro level. In chapter 4, current product development strategies are presented and discussed, from the perspective of design reuse. The need and requisites for a continuous design reuse management are presented. In chapter 5, the main results of this thesis, a framework for continuous reuse management supported by an option-based reuse approach, is presented. In chapter 6, the research and the results are discussed and evaluated from a scientific viewpoint.

2 Research approach

In the previous chapter, we have discussed the problem in industry that this study aims to alleviate: *the problem of missed reuse opportunities over consecutive products*. For practical reasons, this study has to delimit itself to an amount of research reasonable as a PhD work. In this chapter, we present the scope for the research by formulating our research questions, delimitations and expected contributions. In addition the used research method is described.

2.1 Scope

In this section, we formally set the scope of the present study, by formulating our main research question followed by our working questions, delimitations and assumptions.

2.1.1 Research questions

Upon considering the targeted problem, one may begin by asking "why are reuse opportunities being missed?" There is a combination of practical and conceptual reasons for this, and we choose to focus on the conceptual aspects. This which means *problems arising from unknown goals and/or methods and dealing with uncertainty*. In our literature research, we have found no theoretical frameworks covering design reuse decisions under uncertainty, and we now assume that this knowledge gap may be a cause of missed reuse opportunities. In other words, we claim that it is possible that decision makers deciding about reuse under uncertainty could make better decisions if they had such a framework to support their reasoning.

Our main research question is thus:

How can we model the main conceptual approaches to design reuse?

Such a model should explicitly deal with

- uncertainty regarding product requirements and performance of design solutions
    - the framework should support reasoning about reuse despite uncertainties
- drifting contexts including capabilities and market demands
    - the framework should not assume what is known today will be accurate tomorrow
- distribution over time of used resources and consequences of decisions
    - the framework should support weighting short-term vs. long-term aspects

In order to arrive at our framework, we have discerned a number of working questions that we have to answer first, namely:

- What is good design reuse? What is the vision regarding design reuse?
- Where is there most improvement potential for companies?
- Which are the essential design reuse decisions?
- How can the evolution of assortments of products and design solutions be modelled to visualise reuse patterns?

A working assumption is that there sometimes is an apparent conflict between what is optimal considering one-product-at-a-time and what is optimal considering the long-term evolution of the product portfolio. Therefore, we will especially analyse how this conflict is and should be handled affecting design reuse.

2.1.2 Delimitations and assumptions

For practical reasons, this research uses the following delimitations and assumptions.

1
The typical company that this study is concerned with develops and manufactures a product portfolio of high-technological products, for example mechatronic products. This is because we work under the assumption that the engineering costs and the supply chain costs are significant. The company operates under competition, thus forcing it to continuously improve. There is a noticeable grade of innovation between product generations, but these are nonetheless similar to each other, thus giving relevance to the issue of what to reuse.

2
Business networks are not specifically considered.

3
The study does not get into the details of *how* to reuse designs and design for reuse, but rather focuses the decisions of *whether* to reuse and whether to invest for future reuse in different circumstances. While we here focus on the design of products, the reader should keep in mind that ideally the design of products should be done in parallel with the design of other company assets such as the supply chain and the functional organisation. (Design reuse is often an *enabler* for reuse of other company assets.) The study regards the following factors as given:

- product development budget, limited resources
- company and project organisation
- production and logistics infrastructure
- market demand, characteristics, position of company in market (not considering marketing)

4
The question of allocation of responsibility for design solutions (‘ownership’) and other assets is not specifically considered. The thesis will only discuss the degree of strategic importance of different decisions, but not specifically *who* should make them. This delimitation is because of time limits of the study. It appears that the question of ownership indeed has a considerable impact on the behaviour of organisations, and therefore it should be considered in addition to the findings of this study.

5
The problem of too much or inappropriate reuse is only indirectly dealt with. This problem appears when solutions that should be replaced are reused. We acknowledge that such a problem is considerable at many companies which fail to be agile and innovative enough to prevent their product portfolio from becoming stagnated.

6
Branding-driven reuse is not specifically considered here as this kind of reuse is specifically demanded by customers. Instead, this study addresses situations where reuse is optional to the company (at least in theory), and reuse-related decisions are driven by factors such as costs, capabilities and strategy.

2.1.3 Contribution to the research community

Concretely, this thesis aims to contribute to research in engineering design by means of

- an overview of the dynamics of the reuse phenomenon that can be used to diagnose the reuse practices at companies and identify areas of improvement potential
- a conceptual framework to support decision-making in product development, guide high-level reuse strategies and low-level design reuse choices

Hubka and Eder claim that to "*establish a scientific theory of design*", the following aspects need to be covered:

> "*the designer, the activity, the object, the context in which engineering design takes place, and the context of use of the resulting technical system*". [Hubka & Eder 1987, p.123]

Although this thesis does not aspire to establish a theory, we use this list of aspects as a 'checklist' to cover our issue thoroughly. The thesis covers the *designer* by analysing how socio-cognitive and organisational aspects affect his/her reuse decisions. The *activity* of reusing is analysed from the viewpoint of its essential decisions. The *object* of reuse is analysed in the section about 'artefact view', where the general properties of reusable solutions is investigated. The engineering *design context* is studied with focus on knowledge management tools. The *context of use* of a reused solution is considered as how it contributes to (reduce) the costs of the products in which it is reused (we do not consider product use by customer).

2.2 Research method

The research method of this thesis is basically the following.

1) **To clarify and formulate goals** for design reuse and this research by visiting manufacturing companies and performing in-depth interviews, and study existing empirical research in literature. The result of this phase is the problem formulation, i.e. answering '*what can be done better in industry that this study can contribute to?*' This includes areas of improvement potential in industry and academic knowledge gaps, which has been presented in chapter 1.
2) **To analyse the phenomenon** of reuse by an extensive study of literature, from several viewpoints (Figure 3), in order to identify the mechanisms behind design reuse, and map available theoretical and practical support, or lack thereof. The literature used as a base for this phase, which is found in chapter 3, spans from theoretical to empirical. The several viewpoints used to analyse the phenomenon represent different bodies of knowledge of different scientific schools, that complement each other to give a more a balanced understanding of design reuse.
3) **To develop a framework**, combining conceptual tools for different areas and *adapting* and *integrating* them to be useful as decision support for design reuse. This phase addresses the previously identified knowledge gaps, and is based mainly on inductive and abductive reasoning. The result of this phase is intended as the main original contribution of this PhD-thesis.
4) **To evaluate results** and predict implications. The results from this thesis are for practical reasons not tested in real cases (a full scale test would imply experimenting with the product portfolio over many years). Instead, validation of results is done through a check of internal consistency, an evaluation whether the addressed problems are covered, and an argumentation of its usability in practice.

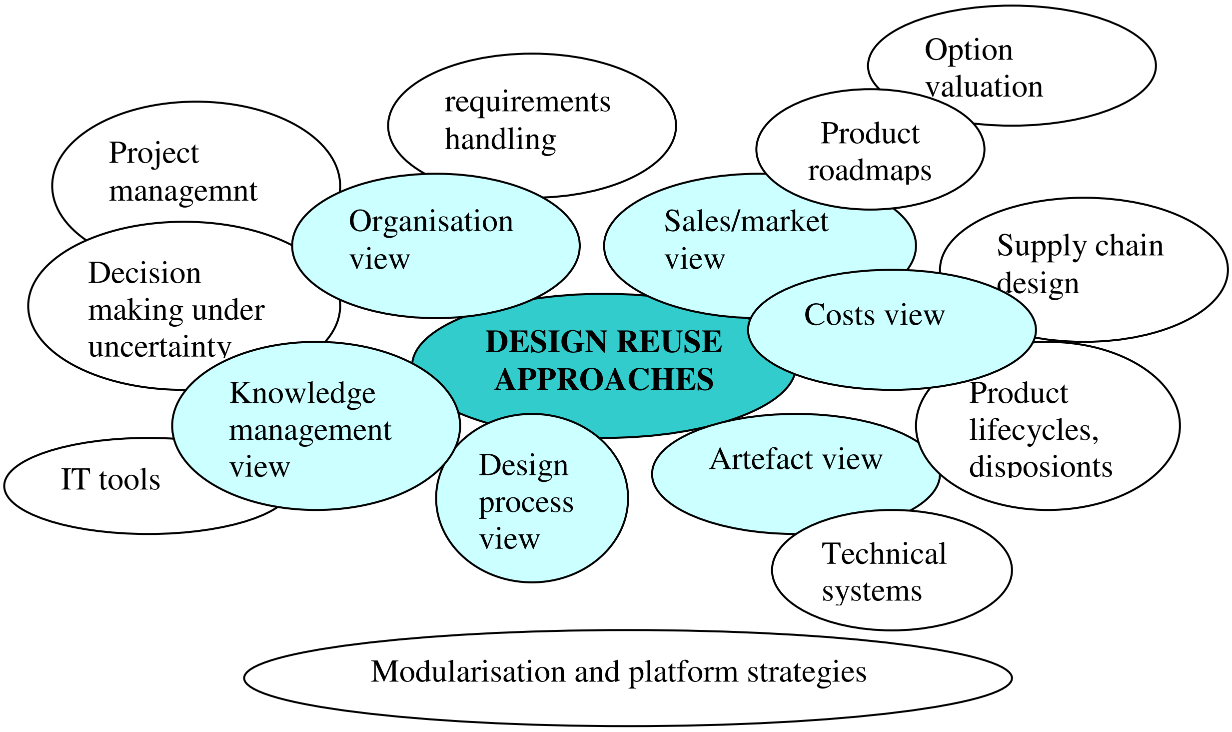


Figure 3: Main topics of the chosen literature references

2.2.1 Research activities

Besides the above mentioned literature study, knowledge acquisition is done through the following activities.

Interviews with product development managers and engineers

An early round of interviews was performed to guide the development of a theoretical model. The interviewed people were all involved in the designing of successive products. The interviews lasted 1 - 1,5 hours each and were performed with one person at a time, except for one interview that included two persons. The interviews were recorded and transcribed. In these interviews, the informants where asked about their perception of the dynamics and context behind design reuse. It was not the intention to evaluate how well these perceptions correspond to 'objective' realities, or to evaluate the companies' performance. Rather, the aim was that these interviews would highlight the important questions from the product designer's point of view, and areas in need of further research. [Jensen & Nilsson 2005, Nilsson 2006a]

Other activities

Visit to DTU, Denmark
Attendance to PhD-dissertations
Conferences:

- NordDesign2004 Tampere
- ICED05 Melbourne
- P2005-conference, The Research Council of Norway, 9-10 feb 2006
- Design2006 Dubrovnik
- NordPLM06, Chalmers, Göteborg
- NordDesign2006, Reykjavik

Summer School on Engineering Design Research

In-depth discussions with engineers in industry (mainly from the automotive and the embedded software industry).

2.2.2 Type of research

Blessing [2005] proposes a Design Research Methodology based on the research stages in the figure below.

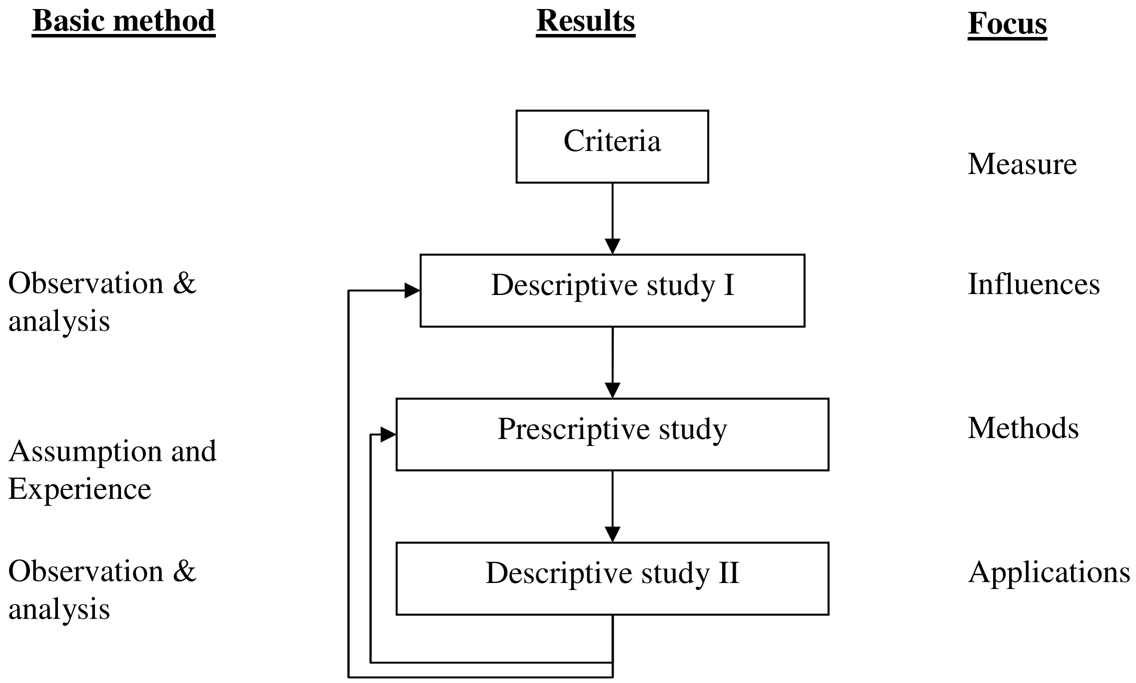


Figure 4: Design Research Methodology [Blessing 2005]

The different stages can be described as follows [Blessing 2005]:

- The Criteria Definition Stage is about finding probable links between the research problem and success. Each of these links and assumptions are compared against literature, in order to know to which degree these have been studied and accepted by the research community. This way, a network of links is formulated, and the links that are to be studied are identified including observable indicators and success criteria.
- The Descriptive Study I is about identifying and understanding the factors that influence the criteria and to provide ground for development of design support and its further evaluation.
- The Prescriptive Study stage is about developing design support based on the Descriptive Study I results and evaluating its internal consistency.
- The Descriptive Study II is about finding out whether the proposed design support can be used in the intended situation and whether it fulfils its purpose (e.g. increase design efficiency).

Based on the above mentioned research methodology, we can identify the following types of research [Blessing 2005]:

Table 1: Types of research according to DRM [Blessing 2005]; type of this research highlighted

CRITERIA FORMULATION	DESCRIPTIVE STUDY I	PRESCRIPTIVE STUDY	DESCRIPTIVE STUDY II	
Review →	→ Detailed			
Review →	**→ Detailed →**	**→ Initial**		*Type of this PhD research*
Review →	→ Review →	→ Detailed →	→ Initial	
Review →	→ Review →	→ Review → Initial / Detailed ←	→ Detailed ←┘	
Review →	→ Detailed →	→ Detailed →	→ Initial	
Review →	→ Review → └←	→ Detailed → ← └←	→ Detailed ←┘	
Review →	→ Detailed → └←	→ Detailed → ← └←	→ Detailed ←┘	

Types of research following DRM [Blessing 2005]; type of this research highlighted
Our research is mainly a "*Prescriptive study I"*-type of research, because it starts with a review of criteria, then makes a detailed descriptive study and ends with some initial prescriptive results.

2.2.3 Research progression

According to Jørgensen[1], applied research should be based on both a problem base and a theory base [Harlou 2006].

---

[1] Jørgensen, K 1992, Videnskabelige arbejdsparadigmer, Institut for Produktion, Aalborg Universitet, Denmark

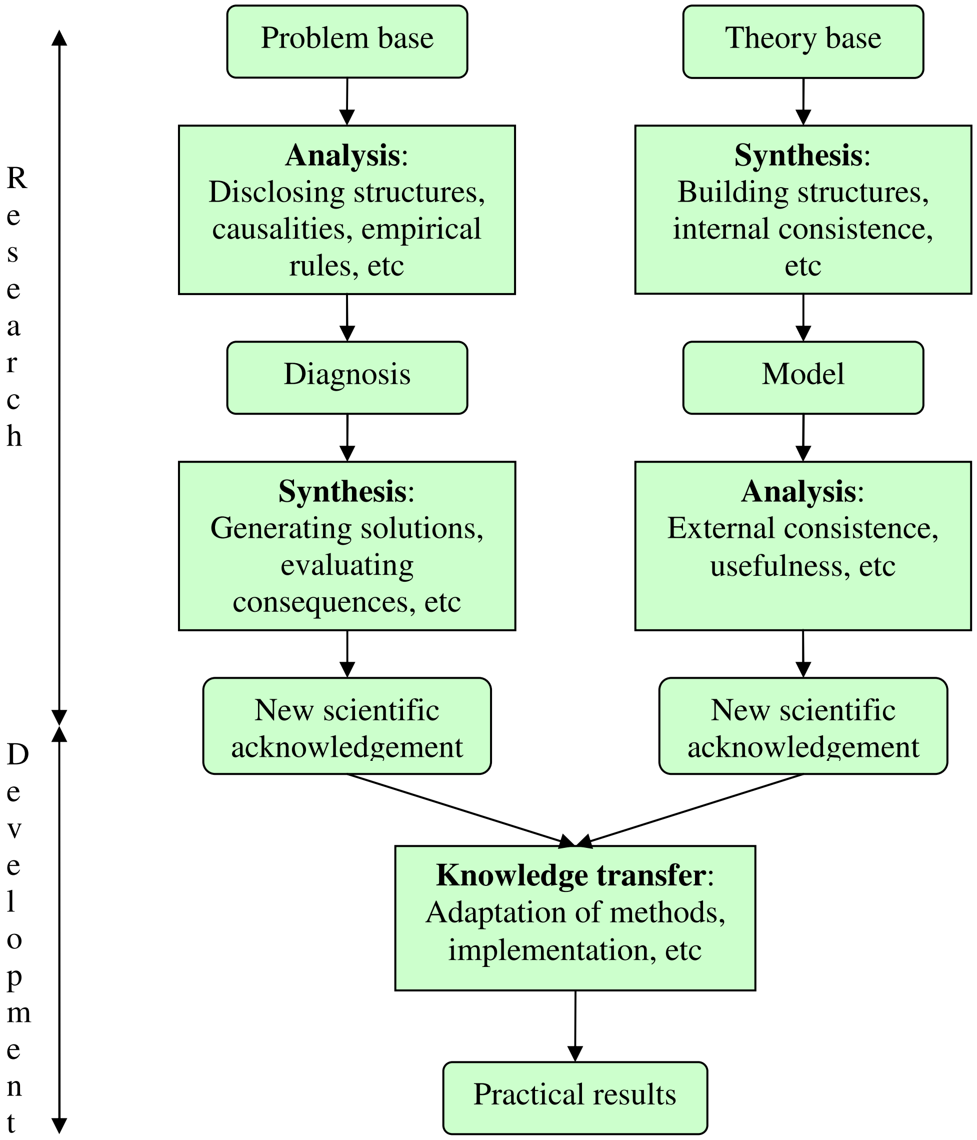


Figure 5: An applied research method, after Jørgensen[1]

Individual research projects, such as this PhD-work, can of course concentrate on portions of the 'chain' in Figure 5, as a partial contribution to the research community. This PhD thesis concentrates on the upper left part of the figure, that is, *Problem-based research*. Theory-based research is performed to provide a terminology and concepts to facilitate the discussion and of the design reuse phenomenon, but it receives less emphasis than the problem-based research. The 'development' part of the research method is because of time constraints left outside this thesis.

2.2.4 Validation method

The issue of how design methodology can be validated is discussed thoroughly by Pedersen et al [2000] and later Seepersad et al [2005]. 'Exact sciences' have traditionally followed a formalistic and quantitative validation, where logical induction and deduction ensure internal consistency. This scientific approach is historically based on the foundationalist/formalist/reductionist school of epistemology, which assumes that "*1) truths (knowledge) are innate and and absolute, 2) that only rational knowledge is valid, and 3) that objectivity exists*" [Seepersad et al 2005, p.5]. However, this approach is problematic for the field of design methodology research that to a degree must be based on subjective statements, because the methods are for and about human beings (except when the research excludes designer-interaction, as in the case of design-optimisation algorithms). Furthermore, design methodology research is expected to be *useful*, which is not guaranteed by internal consistency. Seepersad et al therefore propose a validation to ensure external relevance:

> "*We define scientific knowledge within the filed of engineering design as socially justifiable belief according to the Relativistic School of Epistemology. We do so due to the open nature of design method synthesis, where new knowledge is associated with heuristics and non-precise representations. Thus, Knowledge Validation becomes a process of building confidence in its usefulness with respect to a purpose"* [Seepersad et al 2005, p.8]

To build such confidence in the usefulness of the research, Pedersen et al propose the "validation square", which is based on theoretical and empirical *structural* (qualitative) validation and *performance* (quantitative) validation.

Table 2: The validation square [Persen et al 2000]

THEORETICAL STRUCTURAL VALIDITY (1-2) Correctness of method-constructs, both separately and integrated	THEORETICAL PERFORMANCE VALIDITY (6) Performance of design solutions and method beyond example problems
EMPIRICAL STRUCTURAL VALITIY (3) Appropriateness of example problems used to verify method usefulness	EMPIRICAL PERFORMANCE VALIDITY (4-5) Performance of design solutions and method with respect to example problems

The suggested progression follows the numbers in the square, and is explained as follows:

> "*In (1) we demonstrate that the individual constructs are generally accepted for some limited applications. In (2) we demonstrate the internal consistency of the way the constructs are put together in the method. In (3) we demonstrate that the constructs are applied within their accepted ranges. In (4) we demonstrate the usefulness of the method for some chosen example problems, which in (3) are demonstrated to be appropriate for testing the method. And finally, in (5) we demonstrate that the achieved usefulness is due to applying the method. Based on this we claim generality, i.e., that the method is useful beyond the tested example problems. However, [...] every validation rests ultimately on faith. Hence, the purpose of going through the 'Validation Square' is to present 'circumstantial' evidence to facilitate a leap of faith, i.e., to produce belief in a general usefulness of the method with respect to an articulated purpose.*" [Pedersen et al 2000]

Validation criteria for this thesis

The validation of this work is discussed in the conclusion chapter, from the viewpoints of:

- Internal consistency
- External consistency: is the work based on accepted state of the art in research?

The results of this thesis are exploratory, that is, we propose a theoretical framework on which further research and development can be based to develop practical tools. Therefore, the validation focuses on making a convincing case that the proposed framework is

- internally consistent (is the reasoning logically correct? are the assumptions and the inferences used explained?)
- relevant for the chosen problem to solve
- coherent with widely accepted field knowledge
- meaningful, and useful for further research
    - is it usable to diagnose causes of inefficiencies in real product development cases?
    - does it complement available tools by addressing knowledge gaps?

This corresponds mainly to the upper left area of the validation square, ‘theoretical structural validity’. We return to the actual validation of this thesis in the conclusion chapter.

3 The dynamics of design reuse

In this chapter, the phenomenon of design reuse is studied and described from six viewpoints. Through these viewpoints, different mechanisms that influence the evolution of the product portfolio are exposed. The aim is to analyse the current state in industry and academia and identify areas of improvement potential.

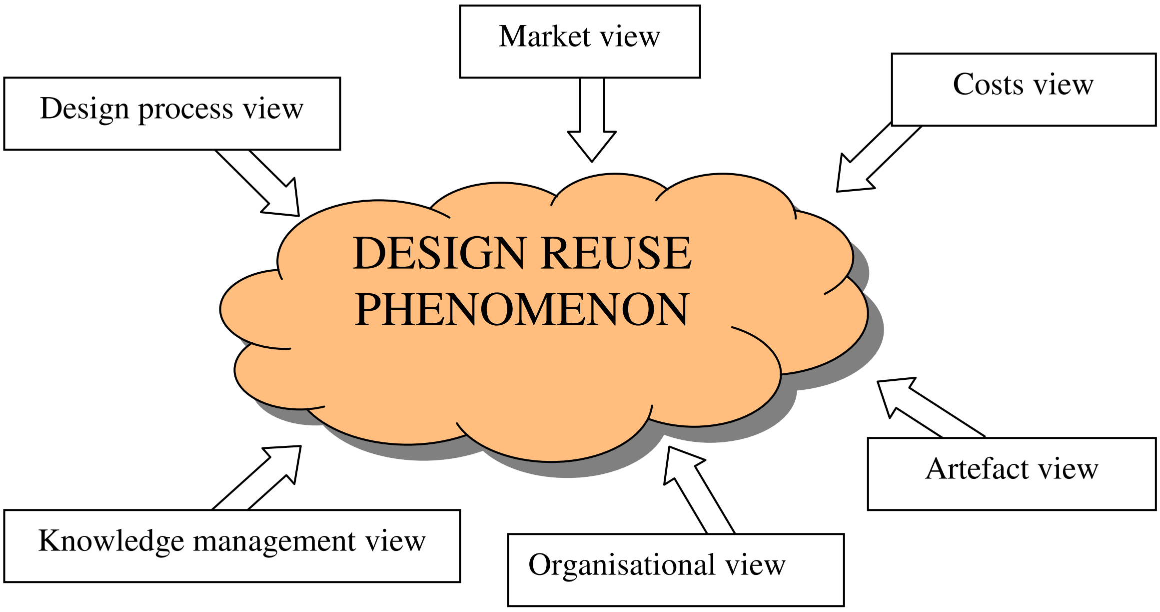


Figure 6: Studying the design reuse phenomenon from six different viewpoints

In the vast literature on product development in general and design reuse in particular that we have we have reviewed, we distinguish three basic ‘requisites’ for reuse, namely: :

- a willingness and empowerment to reuse
- the existence at a given point in time of past design solutions technically suitable for current design problems
- means for these solutions to be transferred, i.e. so that sufficient design information can be searched for and retrieved by the potential reusers

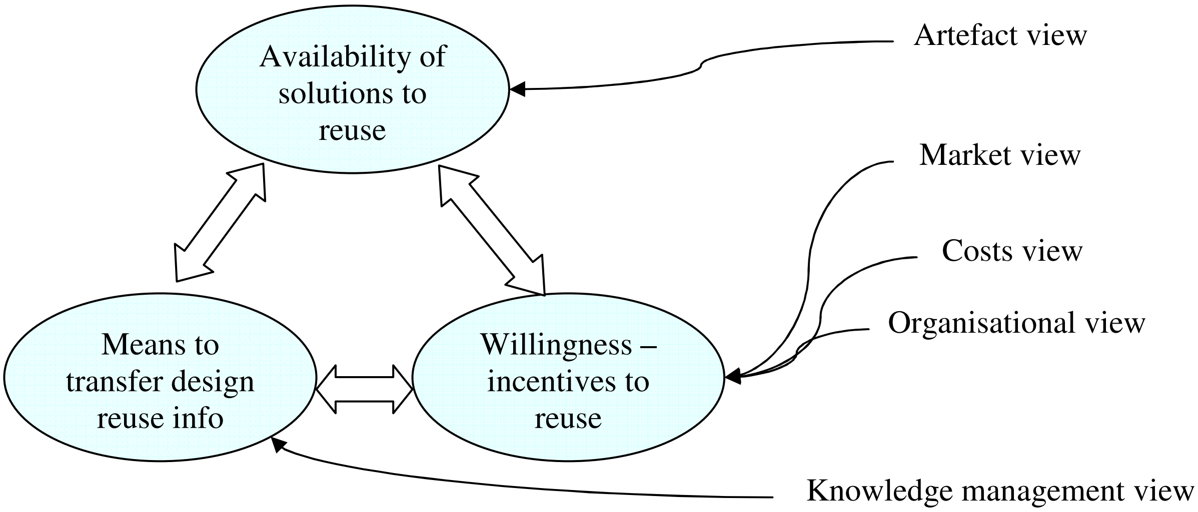


Figure 7: Three enablers for reuse and main relevance to views on reuse

This chapter is divided into six sections, corresponding to the process, market, costs, artefact, knowledge and organisational views, respectively. The section about the Process view provides a framework of activities and flows which we then use in the analysis of the other five views. The 'willingness and empowerment to reuse' mainly stems from the perceived benefits and costs of reuse, and those mechanisms that different stakeholders use to translate perception into action by. The matter is mainly treated in the sections about Market, Costs and Organisational views. The technical characteristics of reusable solutions needed to ensure the 'availability of solutions to reuse' is treated in the Artefact view section. The 'means for transfer of solutions' is treated in the Knowledge management view section.

The table below provides some examples of issues related to the three requisites for reuse, to give the reader a glance of what is to be discussed in this chapter.

Enabler	**Issues**	**Action to improve, examples**
Availability of reusable design solutions	Technical reusability of design solutions (function, structure, match with production and logistics, etc)	Invest in selective development of reuse-friendly solutions
Transfer of solutions	Means to represent, store and locate design information	Document for reuse, store in managed design database
Willingness to reuse	Understanding the benefits of reuse (direct/indirect costs and strategic impact)	Map reuse potential; manage reuse incentives and decision making

3.1 Process view: Reuse in the product development process

In this section, we model design reuse by defining the *activities* and events that constitute the design reuse process and setting them in the context of product development across consecutive product generations. This modelling will in the following sections be used when analysing design reuse from other viewpoints.

3.1.1 Questions

- How can we model the sequence of activities and events of design reuse?
- Which are the essential decisions that determine the reuse process?
- How well do common product design process models fit design reuse?

3.1.2 The design process

To understand the context of design reuse, let us begin by discussing some aspects of the designing process.

Design problems

What are design problems and where do they come from? 'Design problem' refers to a directly or indirectly demanded property of the product that has yet to be designed so that the product can be produced and delivered. A design problem is called *requirement* when articulated purposely. According to common engineering design methodology, designing should begin by analysing the main design problem, (e.g. high-level product requirements), choose a high-level design solution (e.g. a functional structure) that allows its division into sub-problems, which in turn can be cascaded into sub-sub-problems, and so on recursively until the 'leave' problems can be solved. This means the designing progresses from the abstract and rough to the concrete and detailed (e.g. [Pahl & Beitz 1996, Ulrich & Eppinger 2003]. This subdivision allows for a more controllable process and the organisation of the design work into more easily manageable design packages. This process is *recursive* with regard to the creation of design problems, which is especially visible in *modularisation*-based design approaches . Baldwin and Clark claim that: *"The modularisation of a design is essentially a recursive operation – it is a change in the design of a process that changes designs."* [Baldwin & Clark 2000, p.230]

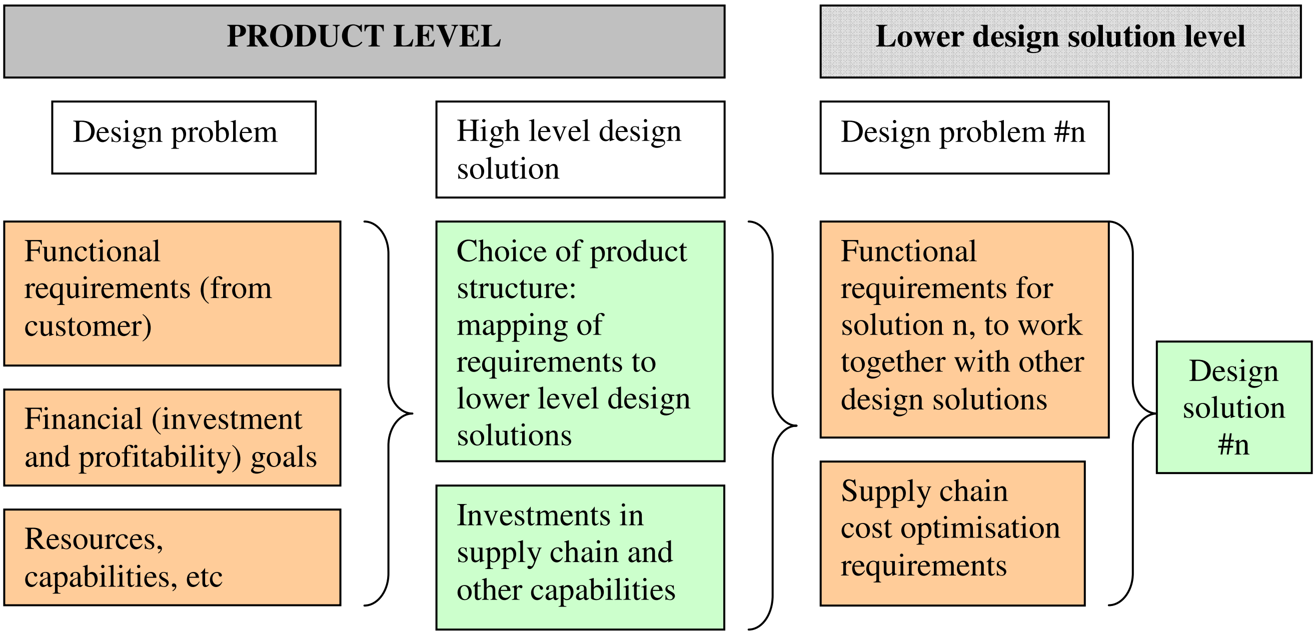


Figure 8: cascading of requirements from higher-level to lower-level design solutions

As suggested by the figure above, a design solution satisfies a number of requirements. Some of these requirements stem directly from customer functional requirements, other requirements stem from other design choices and capabilities and constraints of the company, ideally to satisfy the solution's entire life cycle.

We can call the above recursive approach a top-down design approach, since the higher level design solution is decided upon first, delegating ‘sub-problems’ to be solved by lower level design solutions, and so on. By comparison, a bottom-up design approach would be to begin with lower-level ‘solution fragments’ (for example by assessing previous design solutions for possible reuse), and successively try these in different structural combinations until a satisfactory product structure is found.

Unfortunately, often design solutions have so many dependencies to other solutions and to different company assets that a clean recursive design decomposition is only partially achievable. Of course, the recursive approach can still be, and most often is, beneficial even if it is not ‘clean’ and complete, because it allows for problem solving to get started and interdependencies to become visible at an early stage. But it can be dangerous to ignore dependencies that do exist. When a designer chooses a solution, he/she should be aware of the many possible associated *dispositions*, as explained next.

Dispositions

Olesen defines *disposition* as *“that part of a decision taken within one functional area which affects the type, content, efficiency or progress of activities within other functional areas.”* [Olesen 1992, p.53] For designers, it is important to realise that through design choices, they may be taking decisions about functional areas for which they have no formal responsibility. For example, in a company with poor cross-functional communication, designers may be introducing designs that are impractical from a logistics viewpoint, thus forcing logistics personnel to work inefficiently. Therefore, companies should strive to make the dispositional mechanisms *visible*. This can be aided by design methods addressing the dispositions between designing and other functional areas, such as Design For Manufacturing, Design For Assembly, etc.

In this section, we have highlighted the strive to recursively solve design problems and the existence of dispositions with respect to the development of one product. With respect to design reuse over consecutive products, these mechanisms remain essentially the same, but the added dimensions of time and uncertainty make them more complex to deal with. Designers may have to answer to questions like: How do subdivide a design problem so that the desired resulting design solutions can be reused from the past, or can be reusable in the future? How to know the consequences of a certain design solution regarding manufacturability in a *future* production plant?

Design process models

Some textbook design methodologies propose that we start by identifying and clarifying the design problem, then generate a number of solution concepts, choose the best one and last design the embodiment (concrete details) of the product.

Ullman [2003], for example, propose the following model of design process:

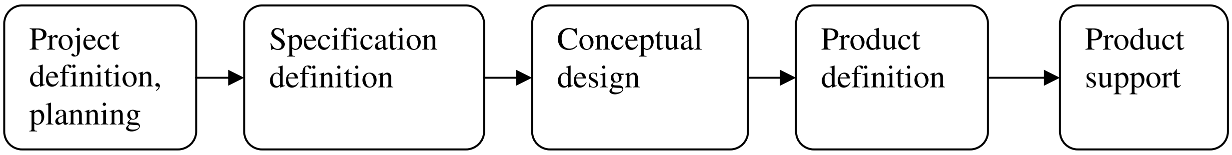


Figure 9: The mechanical design process, adapted from [Ullman 2003]

The following observations can be made from Ullmans model:

- it considers only one project at a time
- does not specifically address whether there are needs for the project to reuse from the past or that the project should provide new reusable designs for the future
- the possible consideration of whether to reuse past solutions (provided this is not explicitly required from the project) would typically be in the "conceptual design" phase, although this is not specifically addressed
- possible consideration of whether to prepare solutions for future reuse is not mentioned, and could fall into any of the first four phases
- the "product support" phase could also include support for future projects that are to reuse designs

Suh [1998] proposes a formal design method which he calls Axiomatic Design Theory. It is based on the notion that designing is the link on *what* we want from the product and *how* we want to achieve it. The *what* stems from customer needs (CN) and should be formulated as a minimum set of functional requirements (FRs) and constraints (Cs), mapped to a set of corresponding design parameters (DPs), which in turn, specify a design to be realised physically by a production process defined as process variables (PVs). *Design* therefore the mapping between these domains. This is explained by Suh as follows (see figure below):

> *"The customer domain is characterized by customer needs or the attributes the customer is looking for in a product or process or systems or materials. In the functional domain, the customer needs are specified in terms of Functional Requirements (FRs) and constraints (Cs). To satisfy the specifed FRs, we conceive design parameters, DPs, in the physical domain. Finally, to produce the product specified in terms of DPs, in the process domain we develop a process that is characterized by process variables, PVs. Many seemingly different design tasks in many different fields can be described in terms of the four design domains, including products, organizations, systems, aterials and software."*
> [Suh 1998 p.204]

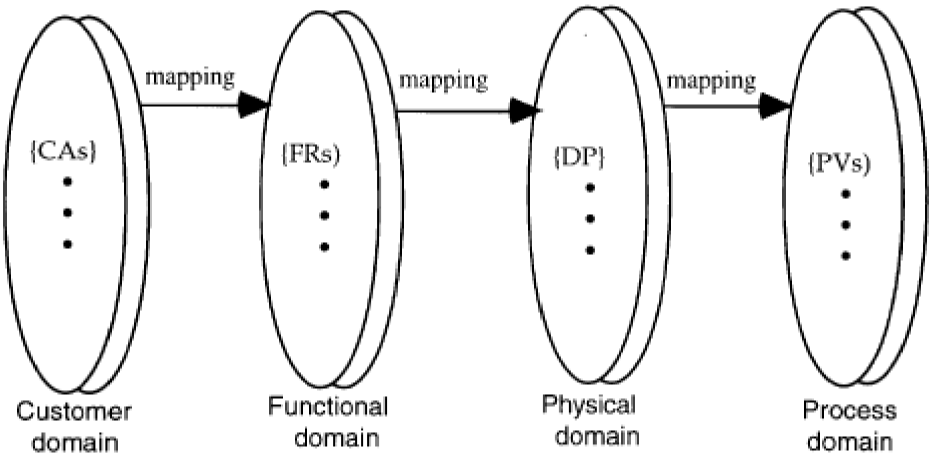


Figure 10: Axiomatic Design models designing as the mapping between four domains [Suh 1998]

Suh proposes a formal notation of the design process. For example, the mapping between FRs and DPs can be expressed:

$\underline{FR} = \underline{\mathbf{A}} * \underline{DP}$

Where A is the design matrix that characterises the particular design. Furthermore, as the name Axiomatic Design implies, Suh bases his approach on two *axioms*:

- The Independence Axiom: "Maintain the independence of the Functional Requirements (FRs)", i.e. maximise *clarity*
- The Information Axiom: "Minimize the information content of the design", i.e. maximise *simplicity*

Axiomatic Design provides a clean formalised way of representing the design process and its goals. However, we note that in reality it appears to be cumbersome to implement, as the theory is based on a formulation of the design task as *uncoupled* functional requirements, i.e. a system free from loops. However such a formalisation in theory is possible, in practice we are used to deal with extremely coupled designs.

Hatchuel and Weil [2003] call Suh's axiomatic design theory a *specification theory*, because it is based on specific properties explicitly required from the product. They propose a different 'Design Theory', which they call C-K theory. It is based on the notion of designing as the cognitive transformation from concepts (C) to knowledge (K). By this, Hatchuel and Weil mean that designers start the designing process with a group of concepts, i.e. vague ideas about how the product can be designed, and that through designing, these concepts are investigated and 'transformed' to knowledge, i.e. a certainty that the particular design problem can be solved in a particular way, or that the concept must be discarded. From knowledge, new concepts can be born to solve newly identified design problems, and so on, see figure below.

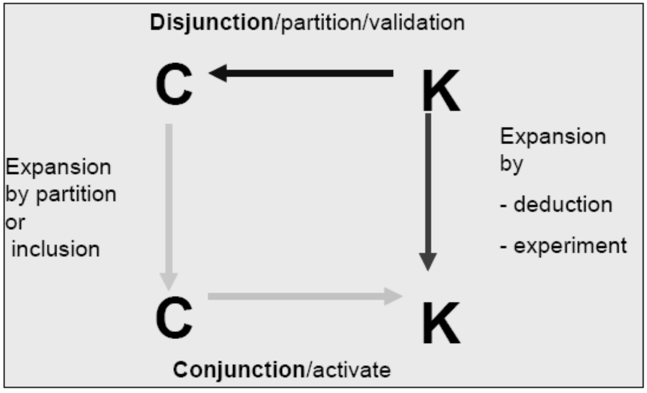


Figure 11: The 'design square' of C-K theory [Hatchuel & Weil 2003]

We can conclude that these design process models do not *explicitly* incorporate design reuse, (although they do not hinder or discourage reuse). One question arises: is it always advantageous that projects start by focusing on the design problem, instead of focusing on already existing solutions? Arguably, the above design process models appear to be influenced by a 'blank piece of paper' mindset, probably with the intention to counter common tendencies by designers to jump to developing solutions without previous understanding of the needs and possibilities, and to counter tendencies to hold on to old solutions uncritically (lack of innovation or excessive/incorrect reuse). So, however this may be a pedagogically appropriate design process model, we should be aware that it does not explicitly encourage reuse, and should be complemented in cases where an increased focus on reuse is wanted.

There are, however, designing methods that do handle reuse, like the one by Otto and Wood [1998] based on redesign, which they summarise as follows (see also figure below):

> *"We start by formulating the customer needs, followed by reverse engineering, creating a functional model through teardowns. The functional model leads to specifications that match the customer needs. Depending upon required redesign scope, new features are possibly conceived, or not. Next, models of the specifications are developed and*

optimized. The new product form is then built and further optimized using designed experiments" [Otto & Wood 1998, p.226]

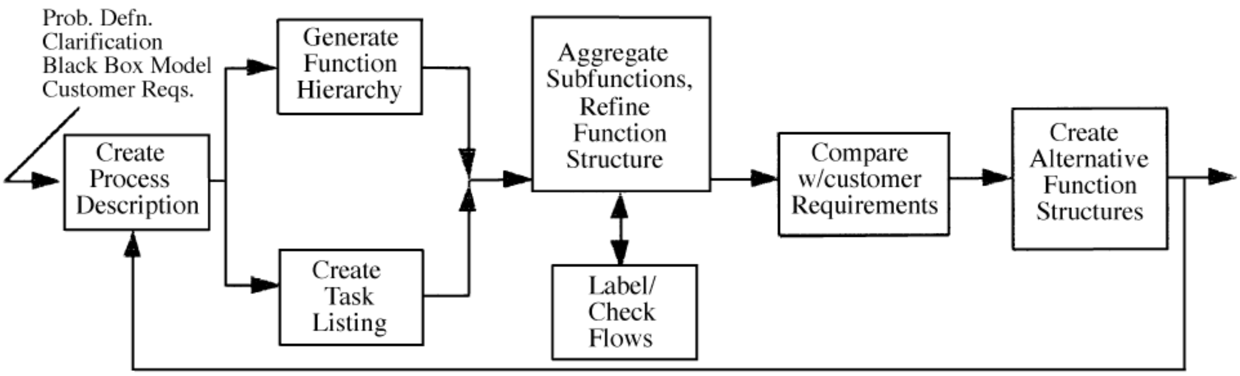


Figure 12: Otto and Wood's [1998] Reverse engineering and redesign method

3.1.3 Previous reuse-related process models

In this section, we review two notable process models in which design reuse has a central role, as a background to the subsequent section where we propose a model tailored for the needs of this research.

Duffy's design reuse model

A design reuse model that specifically focuses on the flow of *knowledge* through reuse has been presented by Smith and Duffy [2001]

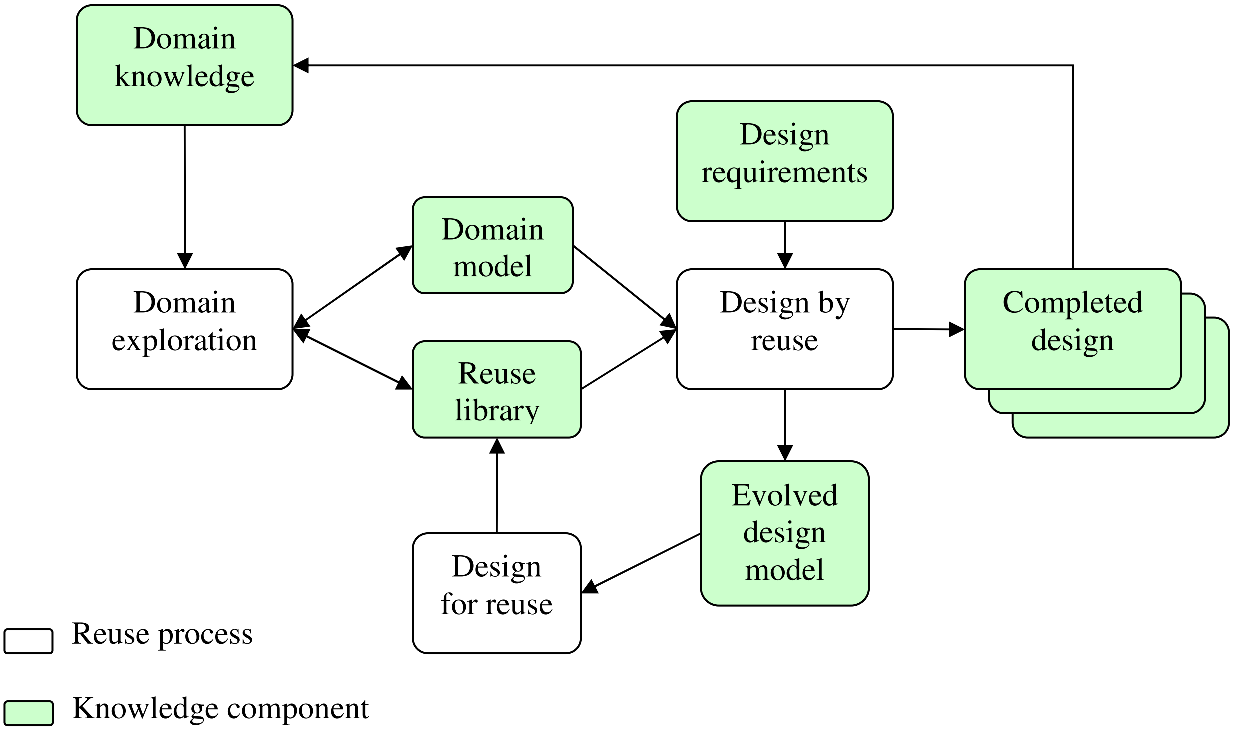


Figure 13: Design reuse model, adapted from [Smith & Duffy 2001]

In the model, the reuse processes are:

- *Design by reuse*: the reuse of past concepts in a new design situation. Reusable "resources" (design solutions) must be available through *domain exploration* or *design for reuse*
- *Domain exploration*: identification, extraction and structuring of reusable fragments from design domain knowledge, for future reuse.
- *Design for reuse*: (during product design) identification, extraction and enhancement of potentially reusable new design knowledge fragments for future reuse.

The knowledge components are

- *Design requirements*
- *Domain knowledge* (e.g. available product information or past design alternatives)
- *Reuse library* (indexed design information repository)
- *Domain model* (the designer's conceptualisation of the current design problem domain)
- *Evolved design model* (a "statement" of an evolved complete or incomplete design, at any abstraction level)
- *Completed design* (a completed "statement" of the finished design solution)

The model can be used to describe a product design situation in the following manner: Designing starts to fulfil a set of design requirements, and a domain model is used together with a reuse library to find solutions. As the design progresses, evolved design models are captured and rationalised for future reuse. After the design is finished, the results and other domain knowledge are analysed through domain exploration, to identify reusable elements.

Smith and Duffy [2001] also relate their reuse model to product structuring. They argue that structuring can be categorised into

- Decomposition: break down the design solution into constitutive elements, often with regard to functions. Decomposition is often necessary during *design by reuse* to find matching elements between a past design solution and the current design problem.
- Configuration: putting together elements to satisfy the requirements on the design, typically useful during *design by reuse*.
- Rationalisation: systematic organisation of product structure-related knowledge, to create a model free from specific quantities. This is typically useful during *domain exploration* and *design for reuse*.

The above model does not explicitly show its linkage with project-based product development processes. Typically, for a given design solution, one could assume that the *design for reuse* or *domain exploration* activity that adds the solution to the design library happens in a project prior to the project where the *design by reuse* happens. We can also observe that the output of *design by reuse* in the model is both 'completed designs' and 'evolved design models'. This dual deliverable (the product to the customers and reusable solutions for future projects) is important to recognise and value as an important output of development projects, as is further argued later in this study.

The above model shows that the reuse process is cyclical when viewing the evolving design knowledge as a whole. But the lifecycles of particular solutions, of course, are not cyclical since each one begins with an original design and ends when the solution is obsolete and no longer reused.

Dynamic Modularisation

A different view of the design process called Dynamic Modularisation, or DYMO, does explicitly consider the development of succession of several products, including reuse and renewal [Riitahuhta 2001].

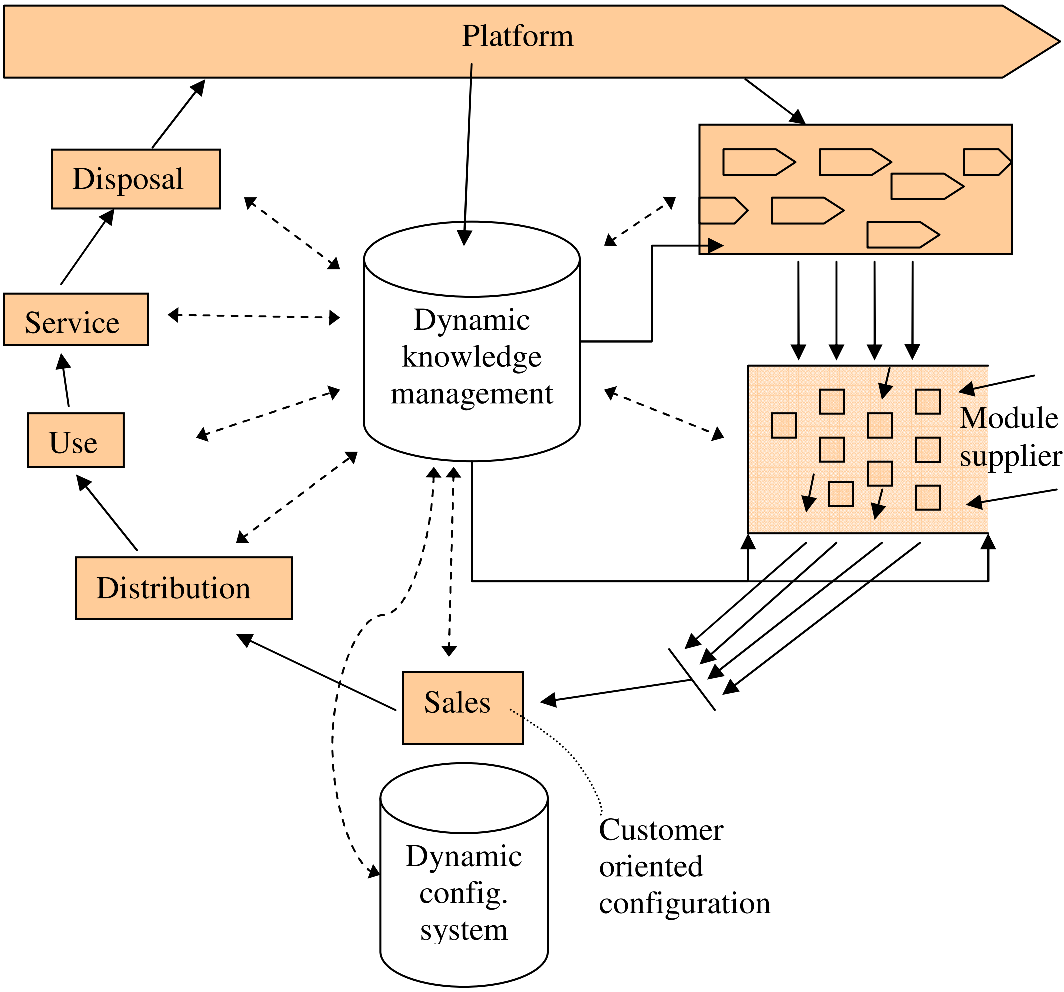


Figure 14: Dynamic Modularisation, adapted from Riitahuhta [2001]

Dynamic Modularisation is defined as follows:

> *"Dynamic Modularisation is the novel modular engineering process, which allows bringing in a dynamic way new more merited modules to the system, and leaving out the old ones. This process is based on the definition of the encapsulation, similarities and the description of interfaces as well as modular management system. All different stakeholders' views should be taken into account; other dimensions will be very similar to those defined for modularisation."* [Riitahuhta 2001 p.16]

Dynamic Modularisation is basically about how to deal with modular products whose modules are gradually renewed over series of new product variants. It stresses the importance of considering the entire product lifecycle upon the design of the product architecture and the selection of modules to renew or remove. This is in our view a sound approach whose essence can be applied also to non-modular products.

3.1.4 A reuse process model based on the flow of design solutions

Below, we propose a reuse-aware model of the ‘flows’ of design solutions from previous development projects to subsequent projects. The main division of the design process is into a ‘producer’ side, where solutions are originally designed and possibly consciously prepared for reuse, and a ‘consumer’ side where past solutions are reused in a new product.

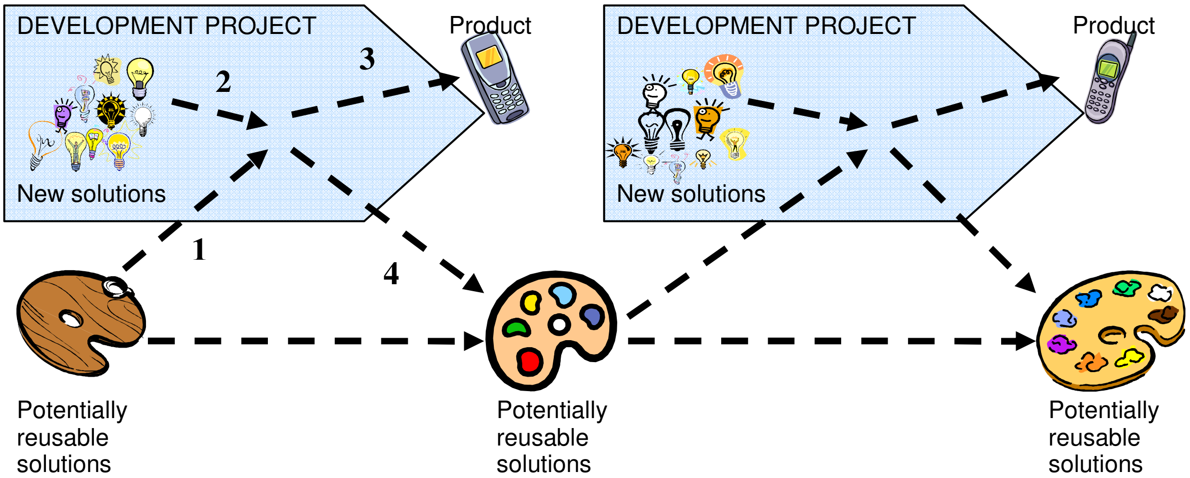


Figure 15: A reuse process model – inter-project flows of design solutions

The figure shows the creation and usage of potentially reusable solutions. In development projects, solutions from a ‘palette’ of past solutions can be used (arrow 1) together with new design solutions (2) to create new products (3). In addition, new potentially reusable solutions can be added for possible future reuse (4). The reuse of past solutions (1) depends on a willingness to reuse, availability of suitable solutions, and an effective means to locate, retrieve and integrate these solutions. The contribution to the assortment of potentially reusable solutions (4) depends on the amount of preparation for reuse. Often, the new product (3) is the only really valuated deliverable of the project, while the contribution to reusable solutions (4) is neglected.

The basic activities of the reuse process

We propose the grouping of the basic reuse-related activities into:

- creation of potentially reusable solutions, including
    - original designing, and
    - designing for reuse, if so decided
- transfer of design solutions
    - preparation for transfer at project of solution origin, if so decided
    - transfer reception, at the project of solution reuse
- reuse
    - reusing as-is (e.g. identical components)
    - partial reuse (reusing with modifications / reusing principles)

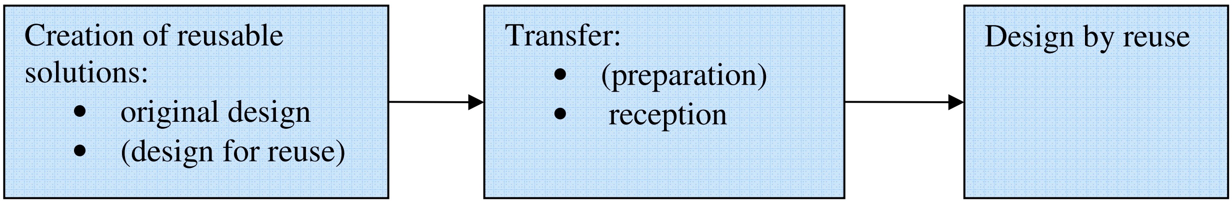


Figure 16: The design reuse lifecycle

Development for reuse
Development for reuse has the aim of preparing for the meeting with the reuser in a future product development project. This imposes additional requirements that one-off solutions do not need to fulfil in the same manner. Preparing a solution for future reuse may include:

- Forecasting future product requirements to specify component reuse requirements
- Designing the solution for reuse, e.g. by
  - 'encapsulating' it with well-defined interfaces
  - increasing robustness so the solution can function in a range of future applications
  - increase design clarity so it is more easy for future reusers to understand, to ensure the solution is correctly reused
- Documenting for reuse

Development for reuse may take place in the scope of

- 'regular' product development projects, where the primary customer is the buyer of the product, and the future reusers of the solutions are 'secondary customers'
- special design-for-reuse projects, such a platform development projects, where the future reusers of the solutions are the primary 'customers'

Transfer for reuse
Transfer of design is the process that transfers the design information pertinent to a solution to be reused from the originating project to the receiving project. This typically includes

- preparation for transfer: capturing and storage of information through documentation at the originating project, and
- transfer reception at the receiving project: location and extraction of the information

Normally, preparation for transfer will greatly facilitate a subsequent transfer reception. Such design information will of course often be useful for other functions in the lifecycle of the design solution as well, such as maintenance and customer support. Furthermore, the effort to rationalise the design information and index it in a 'reuse library', may yield useful insights to the persons involved.

If the preparation for transfer is insufficient, the transfer reception will cost more effort, if possible at all. Then, the potential reuser will have to act as an 'archaeologist' when trying to find and reconstruct the needed information by him/herself, for example through reverse engineering of manufactured components, interviewing the engineers that were involved in the originating project, and so on.

Design by reuse
Given an abstraction/detail level with which a design solution is regarded, we can differentiate between

- reusing as-is (e.g. components and standard interfaces)
- reusing partially (e.g. principles/parameterised design)

When reusing *as-is*, all design parameters of the solution are replicated. When reusing *partially*, some design parameters values are modified. Of course, if one can always choose to conceptually break down a 'partially reused' solution into a part that is reused as-is and another that is not. The reuse choice may even concern a single design parameter.

As is discussed later, often the greatest reuse benefits can be achieved by *partially* reusing a design solution so that the design parameters that are costly to vary are kept constant, while design parameters that are inexpensive to vary are changed to meet product-specific requirements.

3.1.5 The lifecycle of a reused design solution

Before considering individual design solutions, let us first discuss *products*. Products can be said to have a lifecycle, where they 'meet' different life phase systems such as testing, manufacturing, transport, use by the customer, reuse in other settings, retirement, etc [Mortensen 1999]. These meetings are to a degree determined by design choices [Olesen 1992]. It is the entire lifecycle that ultimately determines the long-term success of the product and therefore it should be carefully considered when designing the product.

In contrast to the lifecycle of a product, a reused design solution will 'accompany' two or more different products or *applications* in their meeting with the life phase systems, in addition to 'meeting' the *reuser*. Therefore, a reusable design solution has to satisfy the needs of the different applications in addition to be 'friendly' to the design teams that are to reuse them. Furthermore, the context of a design solution during its lifecycle, consisting of customers, other solutions, knowledge, manufacturing, logistics, etc, usually changes with time and with each application. Thus, the performance of a design solution *relative its context* may change with time, *even if the design solution itself is not modified*.

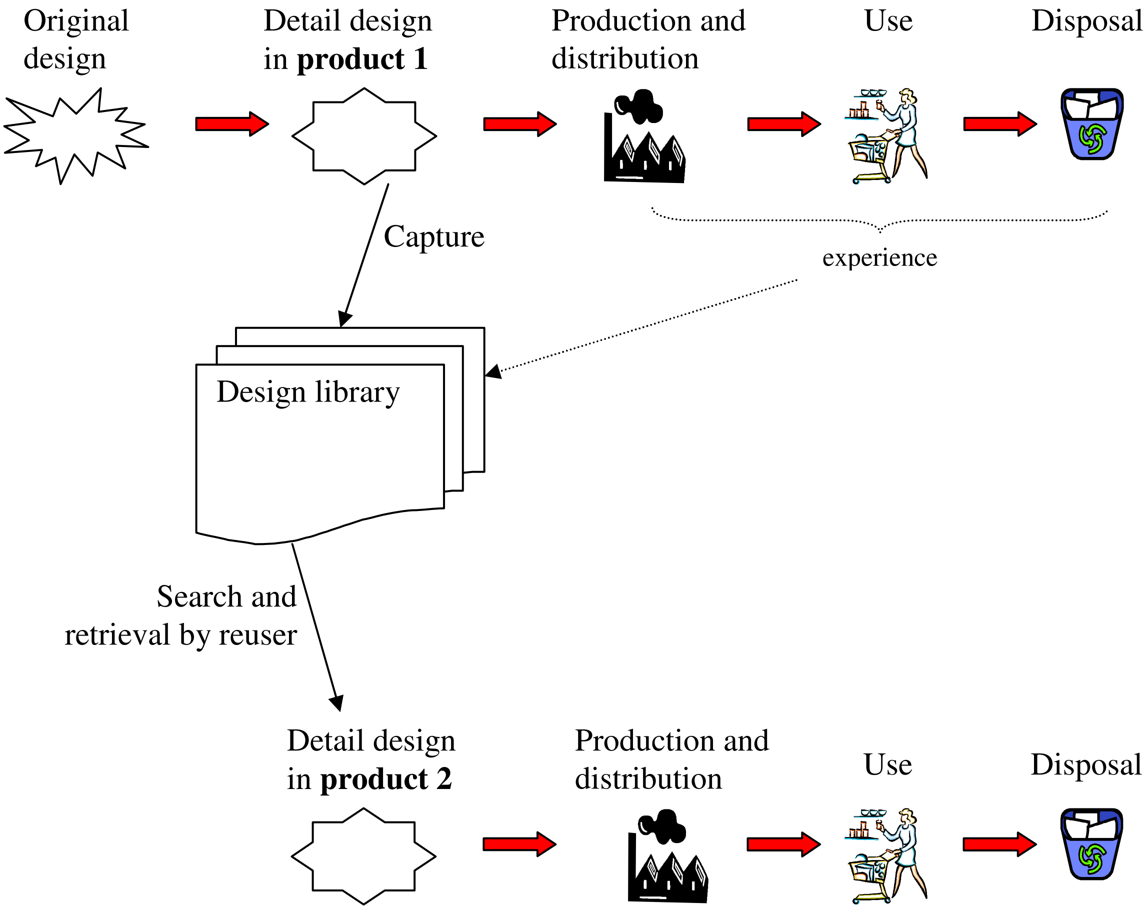


Figure 17: The lifecycle of a reused design solution in two products

3.1.6 The decisions that determine the reuse process

Above, we have modelled the lifecycle of a *reused* design solution into preparation for reuse and execution of reuse. The next step is to formulate a minimum set of decisions that determine *how* individual design solutions flow through the reuse process.

First, it is obvious that the *preparation for reuse* is not indispensable, since some solutions get reused anyway. Therefore, our model must include a decision whether to prepare a solution for reuse or not. Second, our model must obviously include the decision to reuse. Since this study intends to investigate how series of projects are related through design reuse, we distinguish between two types of the decisions to reuse according to *when* it is made, i.e. in the project of solution origin, or in the subsequent project of possible solution reuse. The decision in the project of solution origin is consequently whether to *predetermine future reuse* of the solution in subsequent project(s). The decision in the project of possible reuse is whether to reuse a previous solution (which of course is irrelevant if the solution has been predetermined for reuse). We summarise these decisions with possible choices as follows. In parenthesis, we label the types of reuse processes into ad-hoc, option-based and predetermined, which we introduce in the next section:

In project of solution origin or preparation for reuse:

- Decision to design for reuse: Prepare an existing solution, possibly still at conceptual stage, not fully designed, for future reuse?
    - Do nothing for future reuse (ad hoc reuse), or
    - Prepare for future reuse (predetermined or option-based reuse)
- Predetermine future reuse?
    - Specify mandatory reuse of the solution(predetermined reuse), or
    - Allow reuse to be optional
        - require active decision whether to reuse (option-based reuse), or
        - do not require active decision (ad hoc reuse)

In project of potential solution reuse:

- Reuse previous solution?
    - If reuse is optional, i.e. as-needed by functional requirements
        - Is it worth searching for and evaluating possible previous solutions?
        - If previous solution exists, how does it compare to new potential alternatives?
    - If there is an explicit requirement to reuse solution (predetermined reuse), then this decision is irrelevant.

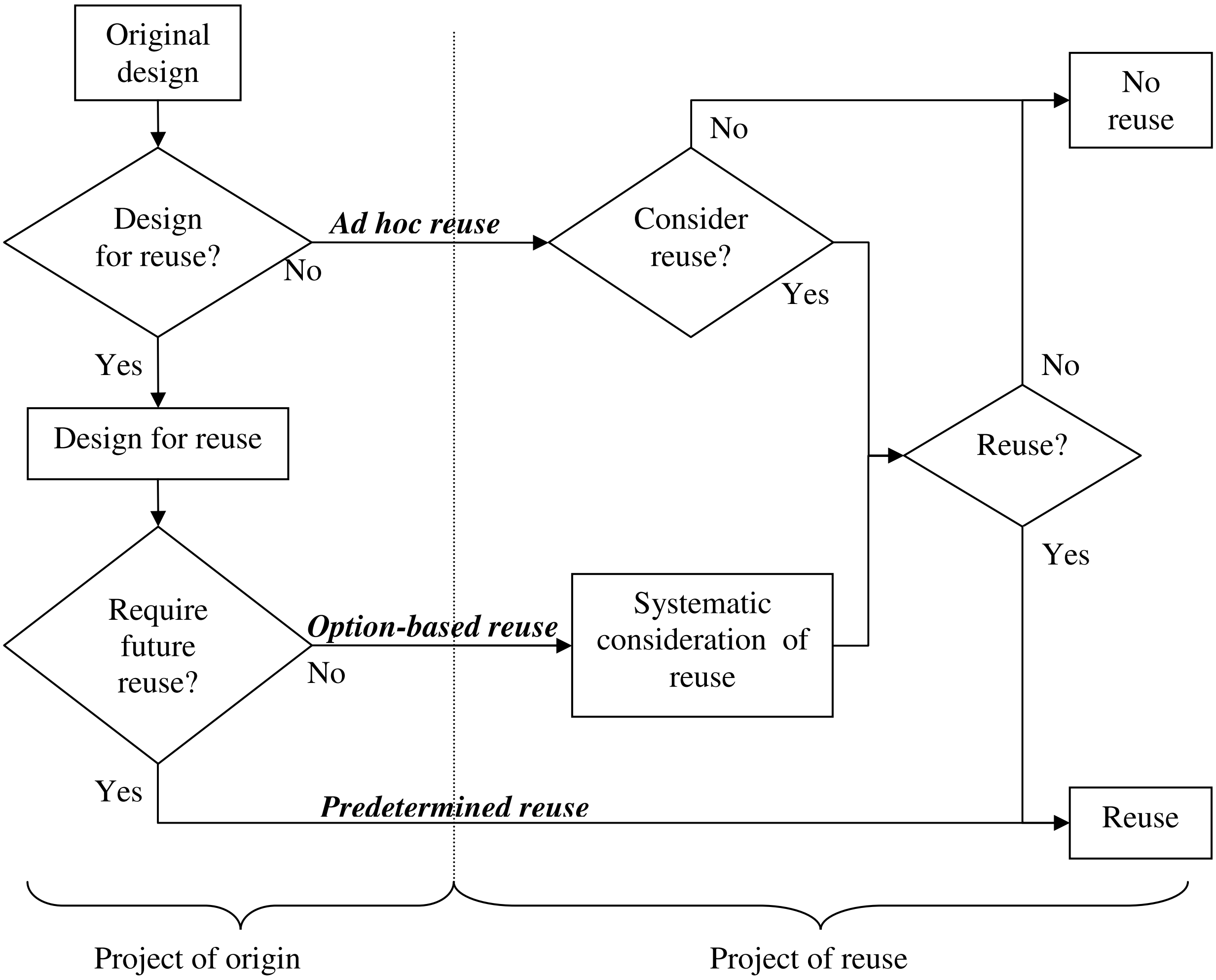


Figure 18: Possible reuse decisions in the life of one design solution

3.1.7 Types of reuse processes

One way to differentiate the reuse processes of differently reused design solutions is by studying the commitment and timing of the *decision to reuse*. Three typical approaches, ‘ad-hoc’, ‘option-based’ and ‘predetermined’ are presented below.

Ad-hoc reuse

When design reuse in products is applied to solve immediate needs (design problems), and without considering future reuse, we can talk of ad hoc reuse. This could be seen as the ‘default’ reuse practice at companies if no proactive efforts are invested in exploiting more benefits from design reuse. Note that ad hoc reuse is not the same as absent reuse, and that companies can apply ad hoc reuse consciously and efficiently, making use of all adequate opportunities to reuse. An effective practice of ad hoc reuse implies willingness to reuse and proficient transfer of solutions (possibility to find previous solutions). Ad hoc reuse may be suitable for companies that develop products that are so mutually different that they rarely can share solutions. Unfortunately, some companies seem to practice ad hoc reuse even though they would benefit from investing in the reusability of selected design solutions.

Option-based reuse cycle

In option-based reuse approaches, future reuse is considered when designing new solutions, but reuse of past solutions is decided on a case-by-case. The aim of the efforts in development for reuse are therefore to provide future projects with the *option* to reuse. Case-by-case reuse approaches fall somewhere in between the ad-hoc and the predetermined reuse approaches.

Predetermined reuse cycle

In predetermined reuse approaches, preferred design solutions are designed and ‘earmarked’ for obligatory future reuse. Because imposing the reuse of a particular solution in a set of future products gives *certainty* the particular reuse will take place, this approach encourages investments to maximise the benefits of reuse. For example, such investments can about higher-volume production machinery to capitalise on economies of scale. Product platforms approaches are based on predetermined reuse, as is explained in the section about product platform strategies below.

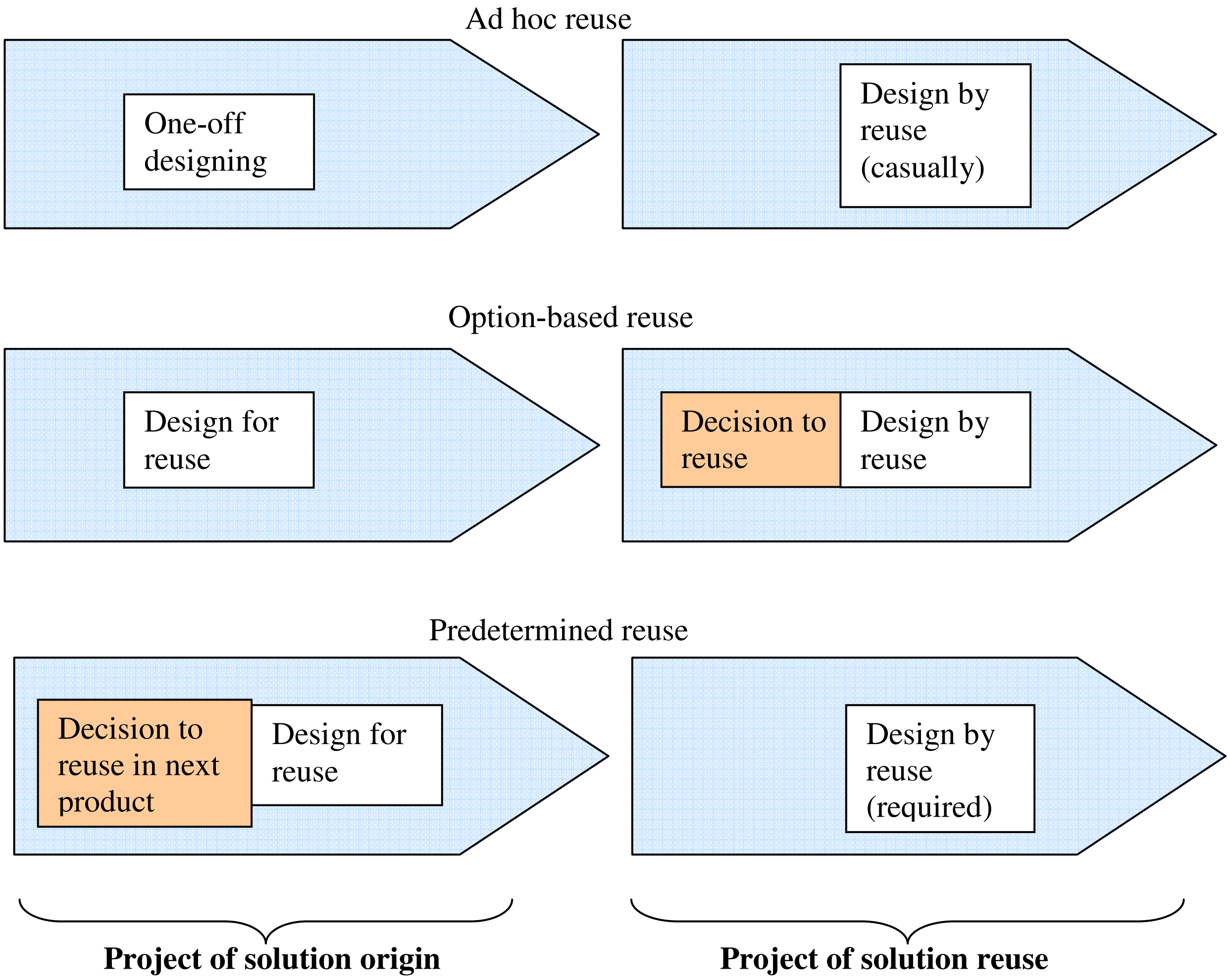


Figure 19: Types of reuse processes according to reuse activities and decisions

Most companies have product portfolios where all three reuse approaches are suitable in different product areas. For example:

- Stable ‘base’ technologies are often predetermined for reuse. This may apply to preferred subsystems, and/or the product structure.
- Technologies that are important from a cost or customer perception point of view but rapidly evolving may be enhanced for possible reuse without predetermining their future reuse (option-based reuse).
- ‘Experimental’ technologies are often left without consideration for future reuse; they may or may not be reused in the future (ad hoc reuse).

Notably, modular product platform development strategies combine predetermined reuse and option-based/ad-hoc reuse. The product structure is designed a-priori and predetermined for reuse in future development projects, while individual modules may be reused or redesigned in subsequent product variant development projects, according to the rules of the platform architecture to meet each variant-specific requirements.

3.1.8 Conclusion

This section has analysed the phenomenon of design reuse from an activities and events point of view and proposed a process model for further analysis. Below are summarised answers to the questions we asked in the beginning of the section.

How can we model the sequence of activities and events of design reuse?
The sequence of activities that contribute to the design reuse process of a given design solution is modelled as three necessary steps:

Creation of reusable solution → Transfer → Reuse

As mentioned earlier, there are many possible variations within each step, the premises and consequences of which are analysed in the following sections.

To capture the essential issues of design reuse, it should be modelled across a sequence of product development projects. The model should be able to show each project's contribution with reusable solutions, as a deliverable apart from the product (except when the project's main deliverable *is* the set of reusable solutions). *Originating projects* add design solutions to the assortment of reusable solutions that are transferred to *receiving projects* for possible reuse.

Which are the essential decisions that determine the reuse process?
We choose to classify the reuse decisions into

- Decision to design for reuse
- Decision to prepare for transfer
- Decision to consider possibility to reuse
- Decision to reuse

As is discussed below in this study, each of these decisions may or may not be made actively/explicitly, and may or may not be made on a case-by-case (solution-by-solution) basis.

How well do common product design process models fit design reuse?
Because design reuse by definition is a multi-product phenomenon, most single-product design process models fail to bring focus to its essential issues.

3.2 Market view: reuse and the competitive environment

In this section, the relationship between reuse and the company environment is analysed, particularly focusing on the market demands in the short term (market responsiveness) and in the long term (strategic positioning).

3.2.1 Questions

- How do market forces determine the need and possibility to reuse?
- How can we classify market/industrial contexts?
- How can we classify product areas according to predictability of reuse needs?

3.2.2 Market demands shaping the need and possibility to reuse

Because of customers' expectations and competition, many companies must address individual customers' requirements closer and closer. This trend is fuelled by technological advancement and improved business practices. Customers' willingness to buy individually tailored products has probably always influenced markets towards increased variety, but this has been balanced by the customers' willingness to accept reduced variety in exchange for lower prices. More and more, companies must *both* offer a high variety of products *and* the low cost of economies of scale (what is often referred to as 'mass customisation'), that is, find solutions that satisfy both variety and commonality needs [Jiao et al 2003].

Identifying just the variety that customers are willing to pay for is naturally a delicate task, which to a high degree has to be addressed with marketing tools combined with cost calculations. Schuh et al [1998] propose a method to select a product portfolio based on "Variant Mode and Effects Analysis" that aims at reducing unnecessary variety, that is, to eliminate product-variant-driven complexity costs that are not justified by market demands.

Arguably, design reuse is most useful for companies in highly competitive markets, where companies must continuously strive to reduce costs and improve market responsiveness with scarce resources. By contrast, reuse may not be such an important tool, compared to other tools as marketing and innovativeness, for companies operating in virgin markets or without serious competition.

Customer demands specifically for/against reuse
In some instances, customers specifically demand design reuse. For example, the customers may want that a previous user interface be preserved because of compatibility to other customer-proprietary components, and convenience of using older spare part inventories. Besides, the customer may desire reuse of certain features because the customer does not want to re-learn how to use the product. There might also be branding reasons, that is, certain features from previous products have become part of the brand image, which is requested by customers.

In other cases, customers may specifically demand *change* of solutions, for example when products having an innovative appeal sell better than products perceived as 'old-fashioned' (despite if the functionality is equivalent).

However, in the general case customers are interested in aspects such as product quality, performance and price, and do not specifically care about reuse per se.

3.2.3 Types of industries

The industrial contexts in which companies operate largely determine the need and possibility to reuse solutions. One important aspect is whether the product development is driven by customers' orders or by expected (but perhaps not articulated) market demands. Another important aspect is how new products relate to the product history of the company. Are new products completely new to the company, or are they based on existing products, either dramatically (mutations) or incrementally (updates)?

Below we mention two other dimensions of the business situation of companies, the type of product differentiation and the industrial clockspeed.

Type of product differentiation
We can classify technology-driven markets according to the nature of customer demands, competition and product differentiation, into:

- Innovation-driven: competition is temporarily non-existent or scarce, success is more determined by marketing issues than by product costs. Radically new, patent-protected products often fall into this category.
- Time-to-market-driven: Evolving technologies under competition, where differentiation is achieved through rapid time-to-market, frequently introducing new products with the aim of technological leadership in the market. An example of such a market is the mobile phone industry, where being first with a new feature on the market means large revenues. The relationship between product development cycle time and commercial success in such markets is for example studied by Meyer and Utterback [1995].
- Cost-driven: defined technologies, product differentiation through price. For example, recordable CDs.

Industry clockspeed

One of the industry properties that have most significant impact on design reuse is the industrial rate of evolution or *clockspeed*. A fast-clockspeed industry has products which have very short market lives, which must be replaced very frequently. The information-entertainment industry is one of the fastest-clockspeed examples. Their products (news, TV-programs and films) have market lives of hours, days, weeks or in some cases months. By contrast, one of the slowest-clockspeed industries is the aircraft manufacturing, whose products may have market lives of decades [Fine 1998].

The product clockspeed is determined both by the demanded frequency of new product introductions and the demanded innovation grade (generational variety) between product generations. With regard to the need and possibility to reuse, it is the demanded innovation grade that matters the most. If the rate of technological change is low (in comparison to the frequency of new product launches), many previous solutions will probably suffice for the needs in new products, i.e. be suitable for reuse. By contrast, if the rate of technological change is high more solutions will become obsolete from one product generation to the next, thus hindering reuse.

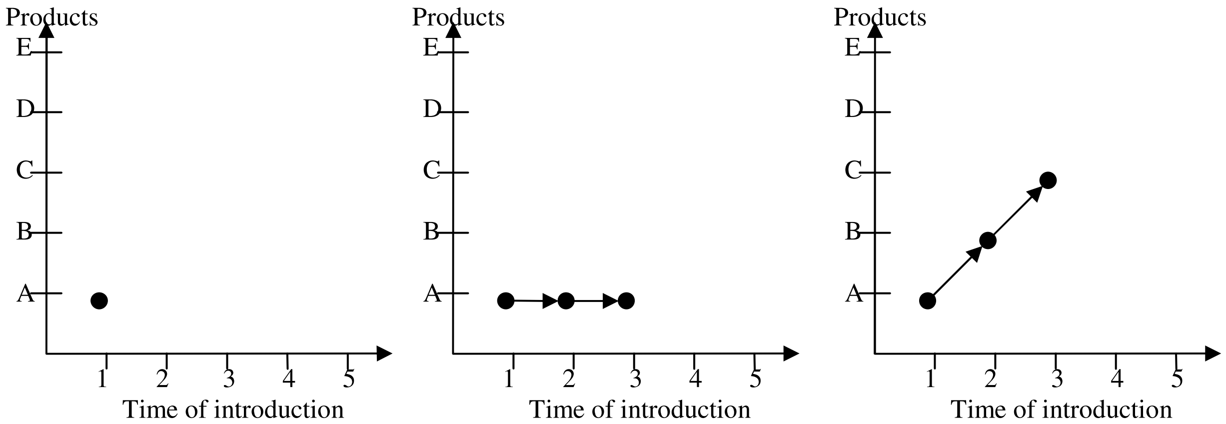
Although we can loosely talk of one prevailing clockspeed for a given industry, it is vital to understand the dynamics of design reuse to realise that at a detailed level, there are relative differences in the clockspeeds of different product features and supply chain assets such as production lines. It may be the case that a given product evolves slowly as seen by the customer, but the production processes that manufacture it evolve rapidly. Within a product, certain features may be stable (stable customer requirements and stable technology) enabling reuse over several product generations. By contrast, other features in the same product may need to change radically from product generation to product generation, so the solutions introduced in one product will be obsolete when the next product generation is to be designed.

3.2.4 Types of product portfolio evolution

Sanderson and Uzumeri [1997] classify the types of product family life cycles into:

- commodity: little change over time and little variety
- variety-intensive: many product variants are introduced and have long life cycles, so there are many variants in use at any given point in time
- change-intensive: product variants are introduced frequently to *replace* previous ones, few variants are in use in parallel at any given point in time
- dynamic: both variety- and change-intensive; many variants introduced and updated over time

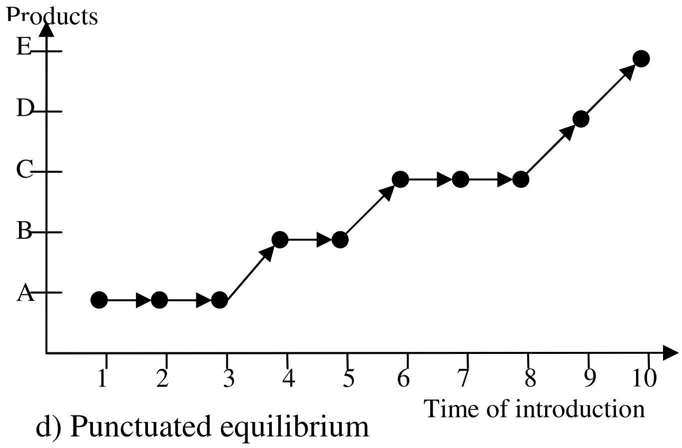
Maier and Fadel [2001] propose a more detailed classification of product families based on number and timing of variants, which they claim is related to the optimum choice of manufacturing paradigm.



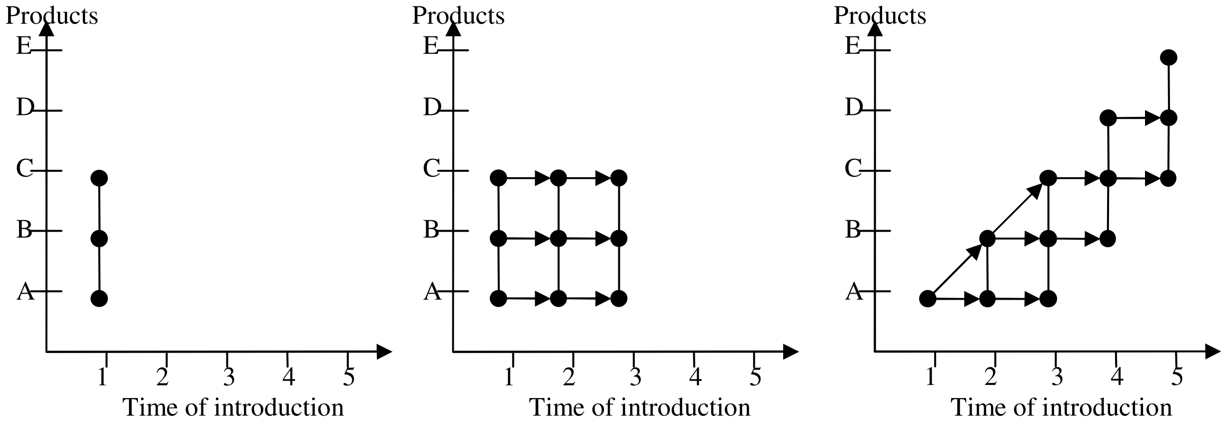
a) Single product design

b) Evolving single product design

c) Mutating single product design



d) Punctuated equilibrium in single product design



e) Concurrent product family design

f) Evolving product family design

g) Mixed evolving/mutating product family design

Figure 20: Types of product family designs [Maier & Fadel 2001]

Maier and Fadel identify seven types product families:

a. Single product design: never modified. Here we find either products so simple they got good enough the first time, or one-off products, customised for a unique situation (e.g. a piece of art).
b. Evolving single product design: one that is periodically updated, but keeps fulfilling the same function.
c. Mutating single product design: is one that is updated into a new product that targets a different market segment than the preceding product, in response to a dynamic market that has given rise to new opportunities.
d. Punctuated equilibrium single product design: are single products that incrementally evolve, which from time to time allows it to mutate to a new market niche.
e. Concurrent product family designs are a set of products that are designed once and never modified.
f. Concurrent evolving product family designs, include a product family that periodically is updated as a group, preserving the functionality of each product.
g. Mixed evolving/mutating product family design: includes various forms of product family evolution, from incremental improvements to mutations to target new market niches. Products may be added and/or removed from this kind of product families.

The product family type that this thesis is most interested about, and which could be called the 'general' case, is the last type, the 'mixed evolving/mutating'. Arguably, it is in this type of company that the design reuse decisions are most complex and in need for conceptual support.

3.2.5 The predictability of future technologies

The predictability of requirements and opportunities in the foreseeable future determines how companies can capitalise on the stable while remaining flexible with regard to the unstable.

Forecasting

Forecasting technologies usually combines prediction of factors that the company has no control over (e.g. a weather forecast) and prediction of factors that the company has influence over. Most forecasting techniques cover the first type, which is called exploratory forecasting. The second type, called normative or goal oriented forecasting, can make the predictions happen by allocating resources. [Kappel 2001] Forecasting at times seem cumbersome to companies, which often forces them to shorten their time windows. Kappel claims that

> *"The painstaking process of gathering data, doing analysis, and applying the templates is a process that customer facing organizations have little patience for. This drives the team to shorten and simplify their time perspective, even though the long term tides may be simpler to predict than the short-term waves. Despite the long-term view promised by roadmapping, roadmaps in practice typically gave serious consideration only to the next product generation (beyond the one currently in development)."* [Kappel 2001 p.47]

Also Meyer and Lehnerd note the difficulties companies have in predicting the technologies that will be available five years into the future, often leading to paralysis because of a constant need for re-planning. Therefore, companies restrict themselves to more certain things, by making minor customisations to existing products. Sometimes, companies may resort to make product plans based exclusively on the state-of-the-art technologies of today, instead of the possible technologies of tomorrow. This 'idealised design'-approach can be complemented by longer-term forecasting:

> *"An idealized design of a new product platform need not be 'perfect,' as its developers will continue to learn about new technologies and to incorporate them into successive generations of the platform. Longer-range technology forecasting is important as a continuous process to refresh the firm's inventory of building blocks; it should not be on the critical path for building critically needed product platforms."* [Meyer & Lehnerd 1997 p.120]

In any case, it is important to remember that most companies deal with a spectrum of technologies ranging from the very stable to the very unpredictable. So often, very much *can* indeed be forecasted even in 'turbulent' industries. For example, some technologies may be bound to *inertia* making them predictable in a foreseeable future. Inertia may be the case when investments have been made in relation to the considered technology (at the own company or at customers), thus making changes unlikely even though new better technology appeared.

In relation to design reuse, one of the key events that companies should try to detect is the *transition* from unpredictable to stable of a technology. When products, parts or manufacturing processes become 'defined', then investments in economies of scale/repetition will be likely to pay off. A defined technology may evolve in a predictable manner, following a *trend*, thus allowing planning accuracy.

The direction and rate of evolution of design solutions

Whether a design solution that is usable in a current product will be reusable in a future product, depends on two main factors:

1. Efficiency: How well the solution meets the design problem, and whether alternative solutions may become available at the time the future product is to be designed
2. Relevance: Whether the current design problem will reoccur in the future product. Perhaps the future product will need to be different because of changed customer needs or a changed product structure? Is the current a product-specific need?

A design solution can be said to be *defined* if there is a fit between technology and needs, and the optimisation criteria is known, even if optimisation can continue for many product generations. We could say that its evolutionary direction is known. A non-defined design solution is a solution whose manner of usage has not been settled among users or is simply not understood by the product planners. Therefore it is not clear in which ways it will need to be optimised in the future, that is, the evolutionary direction of the solution is unknown.

Based on the match between predicted product needs and predicted technologies, product areas can be classified into:

1. Product areas where 'total innovation' is demanded, intentionally ruling out significant design reuse between products (except for 'general' knowledge). Here there is little incentive to invest in reusability e.g. by designing for reuse. This may include 'experimental' design solutions whose future usability is yet unknown.
2. Product areas that can be predicted to evolve, but it is uncertain exactly how. It may be that the solution principle is stable but certain parameters are tuned as gained experience and better tools allows it. The overall optimisation criteria (e.g. decreased weight) and pace of changes may be predictable. This has for example been the case in computer electronics where processing power and memory capacity have increased at an almost constant pace during many years. Here, there is reuse potential, but it can be challenging to decide what and how to reuse.

3. Product areas where future product needs can be predicted and committed to. In this category we find design solutions that can be considered mature, because it is not known how to improve them further and there is no market pressure to improve them further (i.e. they are considered good enough).
4. Finally, there may be product areas where the solutions are known to be becoming obsolete, because there is an alternative that has proven more attractive, or known to be a one-off solution for a design problem that will not reoccur. For example, fax machines, being steadily replaced by other IT solutions.

Of course, many solutions will be ambiguous or hybrids and therefore difficult to categorise. In addition to the technological fit of a design solution, its suitability for reuse will depend on the cost savings by reuse and the costs incurred by reuse, as discussed in the chapter about the costs view.

3.2.6 Conclusion

How do market forces determine the need and possibility to reuse?
Typical customers demand products that are as tailored to their individual needs as possible, but at the same time are inexpensive. Therefore, competition drives the need to create just the right product variety for the customers while *reusing* to achieve commonality and reduce costs.

How can we classify market/industrial contexts?
Reuse is especially important in competitive markets where product costs and time-to-market are vital. We have identified two aspects which determine the need and possibility to reuse: the product differentiation factor and the industrial clockspeed. We identify three stereotypical categories of market situation, according to the demands on the products:

- Total innovation (insignificant competition)
- Evolving under competition (competition driven by time-to-market)
- Mature, cost driven (competition driven by costs)

How can we classify product areas according to predictability of reuse needs?
Product areas or groups of design solutions can be classified into:

- one-off solutions or solutions becoming obsolete (no reuse)
- unpredictable reuse need (experimental, not yet proven solutions)
- evolving solutions (clear potential for reuse, but technically challenging)
- mature solutions (reuse is recommendable)

3.3 Costs view: how reuse affects company costs

This section identifies and analyses the company costs that can be affected by reuse, both positively and negatively.

The expected benefits when reusing design solutions is usually sought in the *avoidance* of different life cycle costs of new solutions [Andreasen 2001][Nilsson 2006b], especially in relation to

- Product development, through avoided designing and testing, potentially freeing resources to innovate more urgent aspects of the products;

- Production and logistics, if the reused designs allow reuse of production resources (less production investment per product), more streamlined production and logistics because of increased commonality, and increased economies of scale [Fisher 1999];
- Internal variety, by avoiding the introduction of parts to the assortment that add to the indirect (complexity-driven) costs of the product portfolio [Blackenfelt 2001].

To achieve these benefits, normally an investment is necessary to make progress in the reuse process.

3.3.1 Questions

- Which economic model of the costs affected by reuse is most useful to our study?
- Which costs are typically affected by reusing a past solution?
- Which are the typical costs of preparing for future reuse?
- What are the challenges when weighting different costs and benefits?

Assumption
In order to give more focus to our study of design reuse, we will in this chapter make one important simplification: *The product requirements (customer-perceived quality), price and launch date are given.*

So the main remaining variables that the company optimise for are
- product development costs
- production and other product costs
- development of capabilities (infrastructure, competences, etc)

This assumption is for example accurate enough regarding many sub-suppliers in the automotive industry, where the company and the customer agree on detailed product specifications and delivery date before development begins.

Therefore, when we discuss design reuse alternatives, we assume these are ‘invisible’ to the customers and therefore mostly a matter of company costs and capabilities. By assuming the launch date is fixed, we assume that the company engages more engineering resources if more product development work is needed, thus increasing the development costs, and vice-versa freeing development resources if less engineering work is needed. In the same manner, we assume that the company engages or frees engineering resources if more or less effort to achieve the required product quality is needed.

Due to the assumption we disregard market effects that delays or quality variations could provoke, and model the effects of reuse on product development efficiency as variations in costs comparable to other company costs.

We are aware that this assumption is not always accurate. First, especially in knowledge-intensive industries, development organisations are not very scalable. Normally there is a limited engineering capacity that cannot be changed rapidly. Second, it is not uncommon that customer-perceived quality can be positively and negatively affected by design reuse. Positive effects may be due to *“the learning and quality improvement associated with increased volume, and because increased production volume may justify higher levels of investment in component development* *and refinement"* [Fisher et al 1999 p.299]. Negative effects may for example stem from compromises on the holistic product properties, such as total weight.

3.3.2 How to account the costs affected by reuse?

There are many ways of accounting for company costs. Normally one prefers economic models that support decision making by modelling the decision variables as inputs, and the effects of interests as output.

Product-based cost models

Traditionally, textbooks on product development have presented a product-based cost models. In these company costs are allocated to products, which supports supervision of which products in the product portfolio that contribute to company profits and which not.

Product-based cost models can also support product design choices. Most product-specific costs become committed during the early stages of product design (conceptual and embodiment phases), making it difficult reduce costs at the later stages. Therefore it is important to try to understand the product cost structure as early as possible, in order to design for minimum cost.

The overall costs of a product can be divided into direct costs and indirect costs, that is, directly allocable to cost carriers (e.g. labour) or not (e.g. building illumination). Furthermore, costs can be classified into variable costs that correlate with production volumes (e.g. material) and fixed costs that do not (e.g. rent of space).

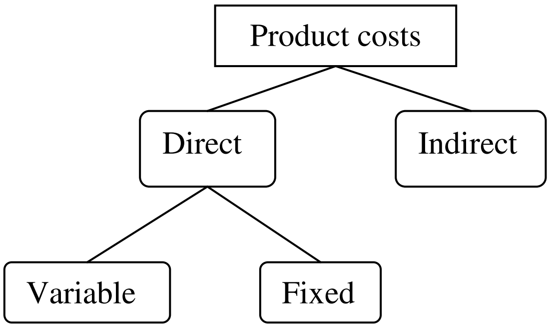


Figure 21: Classification of product costs

Normally, design choices have direct impact on the variable direct costs but not the fixed costs. Indirect costs are per definition difficult to relate to individual design choices, and it is not obvious how design teams should take them into account. Often it is sufficient for designers to optimise with regard to the *variable direct costs*, and simply combine indirect costs with direct costs by means of multiplication factors [Pahl & Beitz 1996].

Product costs can also be classified as corresponding to the different supply chain phases, of which the two largest (for mechanical products) typically are *product development costs* and *production costs*. Production costs include the product-specific variable and fixed costs of material, manufacturing labour, tooling, fixtures and other infrastructure.

Fisher et al [1999] propose a classification of the product-related costs into:

- the investments for new products
    - product development costs (design and test of new components)

    - fixed costs of production (tooling for new components)
- variable costs of production (economies of scale vs. over-design)
- system costs of production, distribution and after-sales support
    - quality assurance, procurement, spare-parts inventory (driven by number of unique parts in the parts in the production and distribution system)

For the purposes of product variant optimisation, Fujita proposes the further subdivision of the *variable* costs into:

- *"Costs dependent on production volume: This mainly concerns material cost, fabrication cost, assembly cost and so forth... The learning effects in fabrication and assembly influence to reduce this category of cost... What is more important, the commonalisation of modules for different products causes excess cost per unit due to over-specification, which is counted as a disadvantage of product variety design."*
- *"Costs dependent on number of product and module kinds: This mainly concern design costs, facility costs, etc., which are usually counted as fixed cost for a single product."* [Fujita 2006, p.192]

Although Fujita acknowledges hidden indirect costs have influence on product variety cost, for practical purposes his optimisation model treats these costs as an insensitive/uncontrollable category.

Fiore proposes accounting for the marginal cost of introducing a new part in a particular company (see figure below). This should help the company be aware when it is profitable to introducing a new part.

> *"To realize a profit, the monetary return from selling the part to the customer must exceed the overall costs the company incurs by creating and maintaining the part itself... For some companies, the carrying cost alone for simply maintaining a part number in its system runs between $2000 and $3000 annually. Factoring in the non-recurring development cost as well as the manufacturing cost, the overall company expense of creating and maintaining parts is substantial."* [Fiore 2005, p92-93]

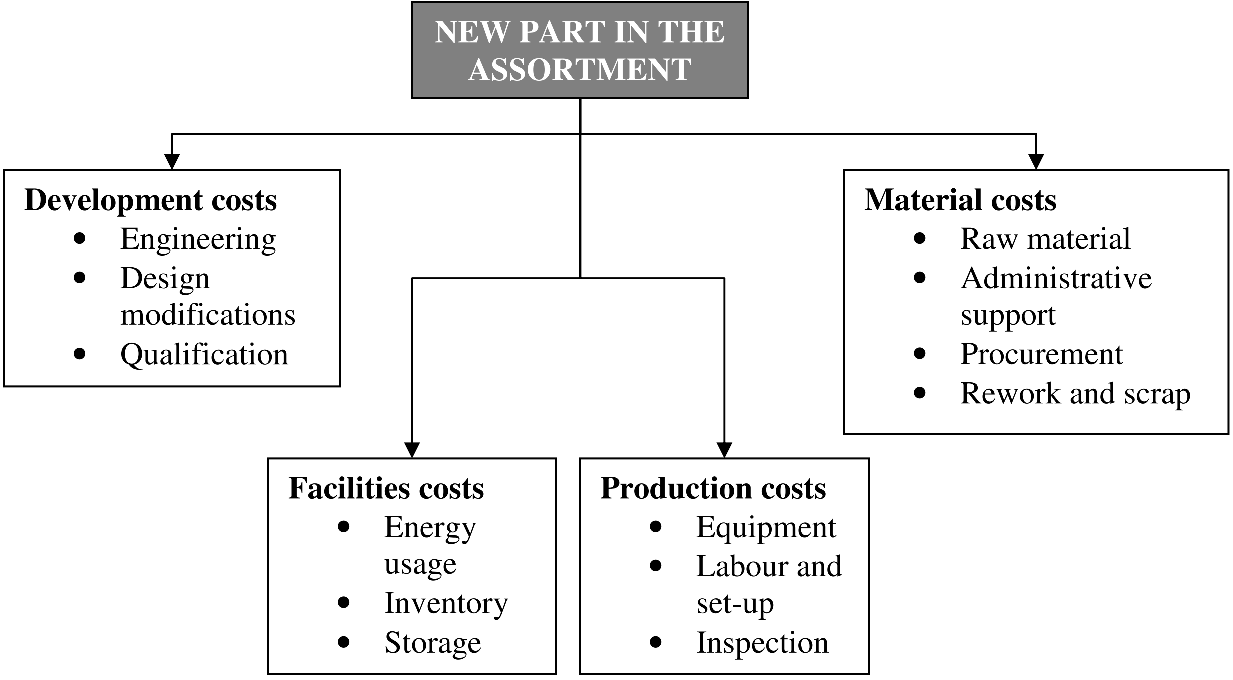


Figure 22: Cost impact of a new part [Fiore 2005]

Activity based costing

Another useful model for making cost estimations is *activity based costing*, which can aid in tracing the costs of development, production and distribution to individual design solutions, as they usually share a considerable amount of indirect costs. Activity based costing is based on the following four steps [Fiore 2005]:

- Resource-consuming activities are identified and costs are allocated to them.
- The activities are studied to find their respective cost drivers.
- Each cost driver or transaction is assigned a calculated cost rate.
- Product costs are computed as the costs driver rate multiplied by the number of cost driver units used by the product.

We refer to Tornberg et al [2002] for a case study exemplifying activity based costing used to design in a cost-conscious manner.

Reuse-specific economic models

To support reuse-related decision making, we need a cost model which has the most significant variables as input and the most important outcomes of reuse as output. Lim [1996] has compared 17 economic models of reuse, and found that the most common *variables* used were:

1. Cost to producer to create asset for reuse
2. Number of times the reusable asset is reused
3. Cost to create product/system without reuse
4. Cost to consumer to reuse asset
5. Cost to consumer to create non-reusable version of asset

The most common type of *output* of the models is *savings from reuse.* Most models do not take strategic impact, overhead nor management costs into consideration, and few did incorporate risk assessment.

Schmid [2003] proposes an outstanding integrated cost- and investment model for product family development, which specifically addresses the economies of software reuse. The model is extensive and covers both the concrete situations of product line tradeoffs, and 'higher level' taking into account financial and risk considerations. The first order model links the product development-side and the market-side. According to the model, the variables of the product-development side (which we can compare to the variables above) are:

- schedule
- effort/cost
- product quality
- development risk

On the other hand, the model's market-side variables are:

- product entry timing
- perceived product quality
- product functionality
- pricing
- product competition

We see that Schmid's model covers a larger space than is the ambition of this study, since this study assumes that price, product quality and delivery date are given.

An investment-based model of design reuse economics

Barnes and Bollinger [1991] argue that the reuse issue should be viewed as an investment. The investment in preparation for reuse should be amortised by the benefits of reuse over one or more product generations. The creation of potentially reusable solutions is the *"reuse-producer"* side, and the reuse itself of previous solutions is the *"reuse-consumer"* side. Their choice of wording producer-consumer is intended as an analogy to a market situation. A particular reuse instance is cost-effective when the aggregate savings at the reuse-consumer side are larger than the net costs at the reuse-producer side.

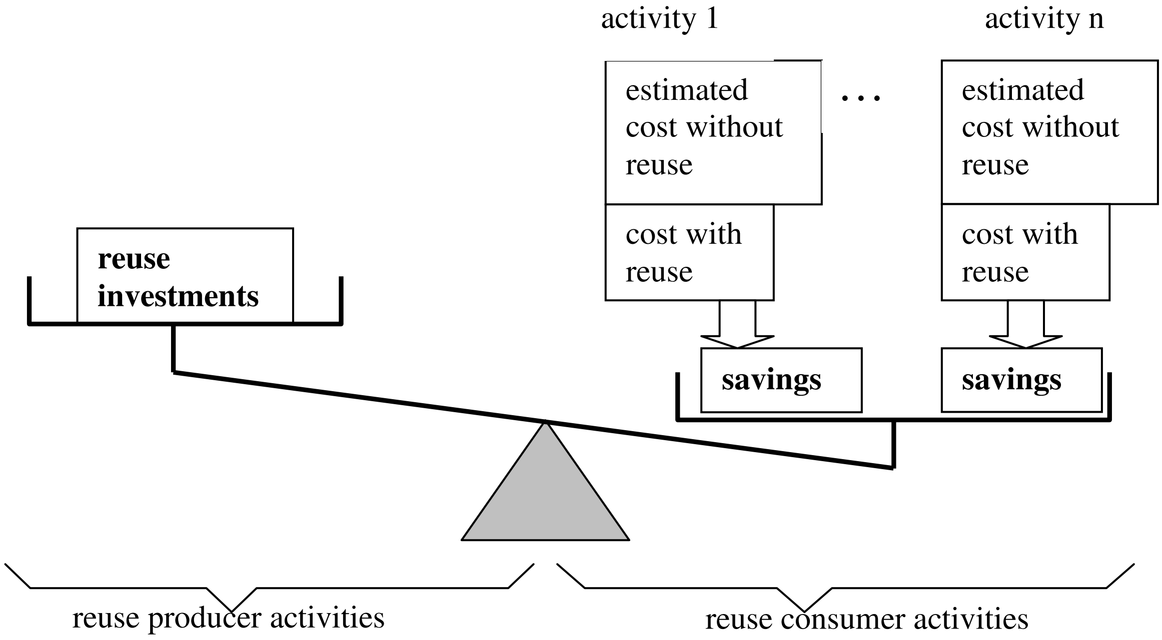


Figure 23: reuse-investment relation (adapted from [Barnes & Bollinger 1991])

In line with Barnes and Bollinger's findings, and for the purpose of studying the specifics of the problem of missed reuse opportunities, we find it useful to categorise the costs in the following way:

1) Costs of reusing a previous solution (excluding design transfer costs) vs. costs of introducing a new one
    a. Product development effort
    b. Supply chain costs (investment in resources, economies of scale vs. over-design)
2) Costs of investments in reusability
    a. Solution-specific (design for reusability, documentation for reuse, over-design)
    b. General, to reduce the average cost of reusing (product data tools, reuse libraries, routines, etc)
3) Costs of design transfer. These costs normally fall both on the reuse-producer side (preparation for transfer) and the reuse-consumer side.

In the following sections, the different categories and their relationships are explained.

3.3.3 Reuse-consumer side: costs affected by reusing past designs

When considering whether to reuse an existing solution or design a new solution, all the significant pro's and con's, both short-term and long-term, should be weighted. The choice may in some cases be obvious, e.g. the reuse benefits are overwhelming, but in other cases it may be subject to a difficult trade-off. If companies do not understand the true supply-chain costs of different design solutions they risk promoting the reuse of high-cost design solutions through reuse programs, which *"can undermine the economic benefits from its implementation."* [Fiore 2005 p.104]

Clark and Fujimoto reflect the variety of issues concerned, when considering reuse of *components* ("off-the-shelf parts") in the automotive industry. They state that reuse of components may mean sharing fixed costs of designing, testing and tooling, among several products. Reuse may even in some cases contribute to decrease development lead-time, *"unless common and carryover parts do not mate well with the new design"*. However, the authors note that these benefits may come at a cost, related to over-design (or under-design), product differentiation and innovation:

> *"When parts not specifically designed for a particular model are used, parameters and functions of the components they are used in may be sub-optimised from a total vehicle perspective... The use of common parts has the potential to increase as well as reduce the lead time and/or engineering hours. Existing component designs that impose inflexible constraints may necessitate additional engineering effort on the rest of the vehicle. Finally, a decision to use an existing component can represent a lost opportunity to introduce a new technology, which may hurt the product's competitiveness in the long run"* [Clark & Fujimoto 1991 p.147-148].

Below, we explore the potential effects of reuse on the aspects of product development costs, over-design and internal variety costs.

Product development costs

Normally, reusing of components (as-is) saves development resources by means of avoided designing and testing. The reused component may require less learning time, because it may be well understood by the design engineers. If it has been used in a previous product on the market, its performance under real use will have been tested, so the technical risk will normally be considerably lower compared to a new solution. Reduction of technical risk and needed testing is an important potential benefit of reusing solutions as-is.

When modifying a reused solution (partial reuse), normally the design solution will have to be fully understood by the reusers. If the reusers are not already familiar with the design solution, learning will normally be dependent on the quality of the preparation for reuse, i.e. how well the solution has been documented and tested for reuse. Often, a modified design solution will need to be fully tested in its new application, but the testing knowledge, procedures and even tooling may be reusable.

The effort needed to reuse a previous solution can be divided into:
Cost of considering if reuse is feasible:

- The number of design issues that a design team can manage is limited. If everything would be to be reassessed regarding potential reuse each time a new product was developed, the loss of focus could make the task costlier. It could be preferable to limit the areas where the potential benefits of reuse are predicted to justify explicit consideration.

Cost of integrating a previous solution to a new product:

- Some design solutions may require more effort to be reused than to be replaced, if they do not easily fit into the rest of the product.

The development costs of reusing may in some cases even outweigh the development savings of reusing, that is, demanding more development effort to reuse than to design a new solution from scratch. This may be the case when it is trivial to design a new solution while an equivalent previous solution is costly to transfer and integrate. (Often, of course, it might be beneficial to reuse anyway, due to other reasons than saving development effort, such as keeping internal variety down.)

Over-design

Component reuse can in some cases increase production costs or reduce technical performance because of over-design. Over-design denotes the design compromise that may be necessary to enable a component to be used both in a high-end and a low-end product [Garud 1995]. The excess capability of a component optimised for a more stringent product *"may incur a unit variable cost penalty relative to the variable costs of unique components designed for each unique product application."* [Fisher 1999, p.299]

Krishnan and Gupta have studied the cost effects of over-design in product platforms, and claim that over-design costs often are considerably larger than the fixed costs of platform development, but that over-design costs may be balanced by other platform-induced cost reductions.

> *"We have noticed in industrial practice that managers tend to be fixated on the fixed costs of development of platforms, even when these costs are negligibly small compared to the gross profit of the product-family. It is more important to note that platforms have a tendency to result in the over-design of low-end products (or the under-design of high-end products) in a multi-product family. This effect may in some cases be outweighed by the beneficial effects platforms have on the product family in general [...] due to a more tightly integrated design and higher volume usage."* [Krishnan & Gupta 2001, p.64]

How much over-design that is acceptable depends on the cost structure of the products. For example, if two products need one particular component type but with differently scaled performance, two main alternatives are feasible. The first alternative is that they share the same component variant, which means the component will be over-dimensioned in one of the products. The second alternative is that the two products use different variants of the component type, each variant scaled optimally for each product's requirements. In the first case, having an over-dimensioned component in one product probably means that some costs increase (e.g. for extra 'unnecessary' material). At the same time, the cost of handling one component variant is probably lower than handling two, e.g. through simpler logistics. These cost sources are of course determined by production volumes.

Internal variety and its impact on supply chain costs

Normally, lower internal variety is beneficial from an overhead cost point of view. Low internal variety can for example allow for fewer machines to manufacture the components with less interruption, allow for simpler logistics, allow the production process to be easier to overview, and/or allow for a more specialized (and less spread) technological knowledge by the personnel.

The costs of internal variety have thoroughly been studied by Franke et al [2002], which also provide a methodology for minimising the cost of assortment complexity. Further research on product variety costs has been done, for example, by Ramdas [2003] and Martin and Ishii [1996]. Internal variety drives costs in different manners in different areas:

- Product development costs increase with the number of unique product structures, features and customisations. A highly complex assortment of solutions and unique components makes it difficult to reapply design processes and automate designing steps.
- Materials costs are driven by the volume of inventories of material waiting to be finished/assembled, which normally increases with the number of manufactured unique parts, and the number of sub-suppliers.
- Manufacturing costs are increased with higher internal variety because it means the opposite of economies of scale. With many unique parts to be manufactured, more special tools, more complex scheduling of facilities, more set-up-time for tools and more manual operations are needed.
- Quality costs are increased due to more unique control and test setups are needed the higher the internal variety is, and learning per design is reduced due to less repetition.

Often, product design literature focus on supply chain cost optimisation by looking at direct component costs one component at a time. Direct costs are normally related to manufacturing labour cost, material cost, transport cost, etc. Excellent designing tools such as Design For Manufacturing, Design For Assembly and other DFX-methodologies provide guidelines for optimal utilisation of materials, assembly sequences, etc, which helps companies save large amounts of avoidable costs.

Unfortunately, often the *indirect* costs receive disproportionably little designing attention, because they are difficult to quantify and link to concrete design choices. Designing to reduce the variety-driven costs implies understanding that commonality is a *relative* property between two or more solutions, that depends on the viewpoint. This is explained by the following examples.

- *Manufacturing commonality*: Manufacturing suffer for example if the machines must be reconfigured offline to produce different components, causing costly line stops. In contrast, if different components can be manufactured with the same machines without extra costs, these components could be said to have manufacturing commonality.
- *Logistics* is affected by the transport and storage of different components. Logistics is very affected by the absolute number of unique components, and holistic properties of the components such as their external dimensions, needed packaging and weight. For example, one component type that is produced in two colours must (from a logistical viewpoint) be considered as two different items. So logistical variety is typically not affected by similarity between components.
- *Operational complexity* is affected by the number and clarity of the interdependencies between the components. If the complexity becomes too high, it gets difficult to overview and make correct decisions about the components (for example know if it is wise to introduce a new component of if there is an existing one that could do the job). That is, once the internal complexity gets out of control it can be very painful to simplify it again. Commonality in this sense may be found between similar components whose relationship to the totality is well defined. For example, having an inventory of a million sizes of screws and nuts would not be regarded as especially complex (although it would be a logistical nightmare).
- *Company knowledge* is affected by variety of different technologies represented. How many different types of technology do the personnel have to understand to perform a

good job at producing and delivering the products? So components or subsystems that are scalable could be said to have commonality from the knowledge point of view.

Indices for measuring commonality are proposed for example by Martin and Ishii [1997] and Jiao and Tseng [2000].

3.3.4 Reuse-producer side: costs of investments in reusability

Solution-specific investments

The preparation for reuse should make the solution more likely to be successfully reusable in the future, with associated cost savings as discussed in the section about reuse-consumer-side costs. Developing solutions for reuse can be considerably more expensive than developing for one product, because of the initial design costs and testing costs [Garud 1995]. These extra costs can be found in different areas*:*

- Development effort costs: Clarification and specification of expected future requirements on the design solution, to know what to design for. Designing the solution to be robust/flexible enough to cover the expected future requirements. Testing the solution more stringently to increase their reliability. The effort of documenting, capturing and otherwise preparing design information for future reuse.
- Over-design costs: If the design solution is to be over-designed to meet future requirements, the current product will have an increased unit cost and performance penalty. It may also be the case that *other* solutions in the current product need to be compromised in some way for the benefit of the reusable solution.

Naturally, often the effort of preparing design solutions for reuse will have a positive effect on quality and knowledge, which may benefit the company even if the reuse does not take place. The issue of *how* to design for reuse is discussed in more detail in the Artefact View section.

Generic investments

Companies may and often should invest in infrastructure and routines to generally improve the effectiveness of reuse. Such efforts can for example aim to:

- reduce the cost of reusing through more efficient design transfer tools (design repositories)
- support reuse-related decision-making with tools to estimate the costs of alternatives (i.e. to reuse a solution or not), especially the costs of variety

The effects of such investments should yield better quality of decisions of what to reuse and maximise the cost savings achieved through reuse. Such gains are naturally often difficult to quantify.

3.3.5 Transfer costs

The transfer cost is the cost of transferring the design information from the project of origin of a given design solution, to the current project considering to reuse it. The transfer costs usually include [Garud 1995]:

1. The effort to represent and update reuse information from the original designers.
2. The effort of the potential reusers to search for and analyse available reusable solutions.
3. The integration of the reused solution information into the current project/product.

The transfer cost is usually negligible in comparison to potential long-term benefits of design reuse. But in some cases, the transfer cost has a short-term penalty on projects that discourages reuse, especially when it is more *predictable* to design a solution from scratch than to spend effort on searching for possible suitable past solutions. This is because part of the transfer cost must be paid even if no suitable reusable solution is found. (Arguably, such companies that can not even afford to consider if there are reusable previous solutions, have considerable improvement potential in the area of knowledge management.) The transfer cost can often be decreased by generic investments in reuse-friendly design information databases, as mentioned before.

3.3.6 Weighting benefits and costs of reuse

The ultimate question when deciding whether a design solution should be reused is, whether the reuse benefits are larger than the costs of reusing. When such is the case, we have *economies of substitution*, which Garud defines as the situation when greater technological progress is achieved by *"substituting certain components of a technological system while reusing others"* than developing the system from scratch [Garud 1995]. The problem is that the different benefits and costs are not easily comparable, since they have different types of impact on business (short-/long-term, cash-flow, development of assets, etc).

One typical trade-off that reuse planners may need to face is the trade-off between over-design and assortment complexity, that is, the costs driven/avoided by product-specific optimisations and the costs driven by increased internal variety. For example, several products may be able to share one component, if this component is over-dimensioned. But if the production volumes are high, this over-design will increase the material costs considerably, making it more convenient to have several material-optimised sizes of the component type.

The issue of over-design is discussed around real cases by Krishnan and Gupta, who argue that companies must weight the 'integration' benefits that follow increased economies of scale with the over-design costs. In their case study, the higher-end product enjoyed a larger decrease in unit cost than the low-end product,

> *"because platform-based development results in an over-design of the lower performance product, offsetting the integration benefits."* Then they conclude that *"if the members of a product family have substantial differences in performance levels, it may be more profitable for the firm to develop the products independently. The platform approach is, in essence, not appropriate for extreme levels of market diversity, and is more beneficial for intermediate levels of diversity."* [Krishnan and Gupta 2001, p53-54, 63]

However, Krishnan and Gupta's findings about the effects of over-design on platform based product development are remarked by Dana [2003], who claims that

> *"Krishnan and Gupta show that platform adoption increases the optimal extent of product differentiation and hence increases the likelihood that two products will be produced rather than one. This remark argues that a change in only one assumption, namely the definition of the over-design costs, reverses that conclusion: Under the different over-design costs, product differentiation will decrease when firms adopt platform-based production, and consequently, simultaneous product introduction is less attractive."*

Consequently, Dana argues, out of their model it is impossible to say "*when the firm will adopt platform production."*

Making economically sensible reuse decisions may be seen as finding a balance between short-term, medium-term and long-term costs. By short-term, we mean costs that affect the current development project; by medium-term, we mean manufacturing and distribution costs of the product; by long-term we mean the impact on the costs of the future products (e.g. internal variety costs). This issue also concerns the *distribution* of development resources and financing over time (should the company be fire-fighting or acting proactively?). Normally, the project budget sets a limit on how much development effort can be spent to reduce future costs.

Meyer et al [1997] present metrics to measure the performance of product platforms. Of these metrics, we find that the *platform efficiency* metric also is applicable to non-platform-related investments in reusable solutions. Platform efficiency intends to answer *"How much did the product cost to develop as a fraction of what was allocated to base platform architecture?"* [Meyer et al 1997, p.93], and is defined as

Platform efficiency = (R&D costs of derivative product) / (R&D costs of platform)

When considering design problems that are repeated in consecutive products, and are 'solved' with the same reusable design solution, we can translate the above metric to:

Reusable solution efficiency = R&D costs of (designing for reuse) / (designing by reuse)

In analogy, cycle time efficiency can be found by comparing the time it takes to solve a given design problem by reusing a previous solution, compared to the time it took to design the reusable solution.

We can conclude that the weighting of all possible benefits against costs of different instances of reuse should ideally be approached with consideration to many aspects of the company's operations and the market's expectations, which are probably impractical to quantify as to automate decision making. Companies can of course develop guidelines to guide low-level decisions in the right direction. At Toyota, the ultimate responsibility for keeping a sound balance between reuse and changes lies at the vehicle programs chief engineers.

> *"Toyota chief engineers have also developed an intuitive 'feel' for how much change in a particular vehicle is just enough. The primary intent is to carry over most of the parts of the vehicle and consider the best utilisation of exiting tooling; only then will a chief engineer consider where and how to introduce new technologies. This is a stark contrast to the 'clean sheet' approach historically employed by NAC and other companies."*
> [Morgan & Liker 2006 p.45]

3.3.7 Conclusion

Which economic model of the costs affected by reuse is most useful to our study?

Design reuse has economic effects that often are difficult to quantify and show in the long term. The benefits of reuse lie mainly in avoided costs of reusing. When these are larger than the costs of preparing for reuse and of reusing, we have economies of substitution. Reuse-related benefits and costs can be of short-term scope, like PD lead time and efforts, and long-term scope, like decreased costs of complexity and provision of reusable solutions for future projects.

The costs affected by reuse affect either the reuse-producer side (preparation for future reuse) or the reuse-consumer side (the actual reuse of a previous solution). Transfer costs are normally borne partially by the reuse-producer and partially by the reuse-consumer side. In reuse unfriendly circumstances, the transfer costs can be considerable and be borne mostly by the reuse-consumer side, thus hindering effective reuse. But if efforts are made to facilitate the representation and transfer of design information, transfer costs can be negligible.

Which costs are typically affected by reusing a past solution?
On the reuse-consumer side, reuse can affect

- project development costs
    - positively, if reusing a solution allows the project to meet the requirements faster
    - negatively, if reusing a solution implies a transfer and integration effort that is larger than the effort that would have been needed to design a new solution
- manufacturing costs
    - positively, if reusing a solution allows increased economies of scale (integration benefits)
    - negatively, if the reused solution (or other solutions) must be technically compromised (over-design)
- complexity costs of the assortment
    - positively, if reusing a solution implies a simplification of operations
    - negatively, if a new solution would have allowed rationalisation

Therefore, costs can both encourage or discourage reuse, depending on the circumstances. For example, it may be deemed disadvantageous in the short-term to reuse a solution (because it is faster to design it from scratch) but advantageous in the long-term because of smaller internal variety. Or, it may be advantageous in the short term to reuse a solution because it saves designing time, but disadvantageous in the long term because it hinders a cost-saving improvement.

Which are the typical costs of preparing for future reuse?
The reuse-producer side costs can be classified into

- generic investments in reusability, which has the aim of improving reuse efficiency in general
- solution-specific investments, which is the extra effort and technical compromise put into one design solution to increase its reusability

What are the challenges when weighting different costs and benefits?
The challenges stem from the fact that different costs impact different areas of business at different times. Companies must themselves find appropriate weightings that suit their strategies regarding the amount and distribution over time of investments, costs, risks and revenues.

3.4 Artefact view: What are the technical characteristics of reusable solutions?

This point of view concerns primarily the designers. From an artefact viewpoint, the central issue is what technical characteristics make design solutions reusable. The artefact viewpoint is more of a ‘snapshot’ or static view (by contrast to the dynamic and evolutionary view presented in the ‘Process view’ section).

3.4.1 Questions

- What do we call a ‘design solution’?
- How can solutions be reused?
- What are the criteria to know if a past solution is reusable?
- How to design for reuse?

3.4.2 Technical concepts

As a background to the concept of ‘design solution’, we first review two other important concepts technical systems and product architectures.

Technical systems

The Theory of Technical Systems [1988] describes the relationships between elements in products and their environments. Hubka and Eder define a ‘technical process system’ in which there are four subsystems

- the product (technical system itself)
- the user or human operator (human system)
- the environmental system
- the meeting between the above subsystems (technical process system)

Furthermore, they define a ‘transformation process’, meaning the transformations that the product gives rise to and are of value to its user (its ‘functionality’); and an ‘organ structure’ which is the way in which transformations are allocated to working elements.

The theory of technical systems has been further developed by Mogens Myrup Andreasen’s Theory of Domains and later the Chromosome Product Model [Mortensen 1999]. The Chromosome model covers the three domains Transformation Domain, Organ Domain and Parts Domain. These are modelled from the following abstractions:

- Chromosome model
  - Constitutive model (technology, organs, parts)
  - Behavioural model
    - ‘soll’ (desired) behaviour
    - ‘ist’ (actual) behaviour (process, functions, tasks)

A design’s *organs* are roughly the functions necessary to fulfil the product’s transformations, whereas the *parts* are the physical units that realise the organs’ functions.

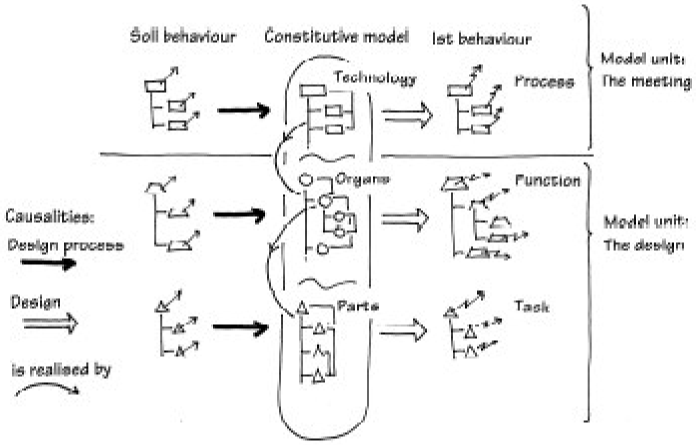


Figure 24: The chromosome product model [Mortensen 1999]

The ability to analyse products at a functional level (organs), and not only implementations (parts) is often vital for efficient designing and can be used to identify product similarity and functional interdependence across several products [McAdams et al 1999]. These dimensions are, in turn, indicators of reuse potential.

Product architecture

A product architecture is defined by Ulrich as "*the scheme by which the function of a product is allocated to physical components*" [Ulrich 1995, p.419]. He proposes a classification of products according to their modularity, from fully modular (one-to-one mapping of functions to components) to fully integral (all functions to the same component). Fixson argues that 'modularity' is a too vague and complex dimension to be operational, and proposes a more specific description of product structure characteristics [Fixson 2004], to link product, process, and supply chain design decisions. This is composed of two 'product architecture assessment dimensions':

- the function-component allocation scheme, and
- the interface characteristics
    - interface type
    - interface reversibility
    - interface standardisation

Design solutions

A product design is composed of a number of design solutions, of arbitrary granularity. In this study, we regard a design solution as a design *decision* made to solve a design problem posed directly by product requirements, or by other design solutions. As we have discussed before, the choice of a higher-level design solution may 'give rise to' a number of lower-level design problems, until all necessary design details have been determined.

A design solution may have one or many designable dimensions or parameters (geometrical dimensions, material, finish, etc). Often it can be useful to make a distinction between solutions representing 'structure' (e.g. interfaces, functionality mapping) and solutions representing 'content' (e.g. components). Design solutions may also correspond to more holistic properties of the products (i.e. indirectly decided by the lower level design solutions), such as aesthetics, overall weight, and supply chain issues such as the choice of production method or supplier.

The dimensions that designers can control directly are actually not many, in the case of mechanical products: form, size, material and surface. Properties are always relative to other assets and can only be controlled *indirectly*. For example, unit cost is a property that is dependent on the cost of materials, of labour, etc; unit weight depends on the form, size and choice of material.

Design solutions may be classified according to the:

- Level of abstraction: from design principles to concrete implementations (e.g. components). One can abstract from many viewpoints, like energy flows and functions.
- Level of detail: from systems to small parts. Normally, there are hierarchies of design solutions, that is, solutions may refer to nested 'sub-solutions'. For example, a high-level solution may be that "the product shall consist of subsystems A and B". At a low level, a solution may be that "the length of screw #123 shall be 12mm".

In order to discover reuse potential at different levels, it may be necessary to make jumps in granularity (zoom in and zoom out). For example, one solution AB may consist of the combination of sub-solutions A and B. In a given situation, the reusability of A could be considered against a potential alternative A', assuming AB is reused. But zooming out, the reusability of AB itself could be considered against an alternative solution D.

Assortments of design solutions
The group of all solutions existing in a company at a particular point in time we call the assortment of design solutions. This assortment has been built up throughout the product history of the company and contains solutions that have been deliberately designed for reuse (generic) and intended one-off solutions (product specific). Naturally, the reusability of existing solutions changes as the company's products evolve technologically. The assortment of solutions normally increases as new products are added to the product history, because old solutions are often kept a relatively long time at a company, in one form or another, before they are completely removed.

Commonality
Commonality is a relational property between two or more items in relation to given commonality criteria that are determined by the viewpoint, e.g. a stage in production and logistics. Therefore, two items can have commonality without being identical. For example, seen from the perspective of a transport mechanism, two items may have commonality if their maximum dimensions allow them to fit. In the task of decreasing the assortment-driven costs of a company, it is important to identify which commonality is really needed at each stage of production and logistics.

It is worth noting that two different components can have commonality from the viewpoint of a certain company asset, if it allows for reuse of that asset. For example, two components that use the same material may allow the reuse of a material supplier, regardless if the designs are based on different working principles.

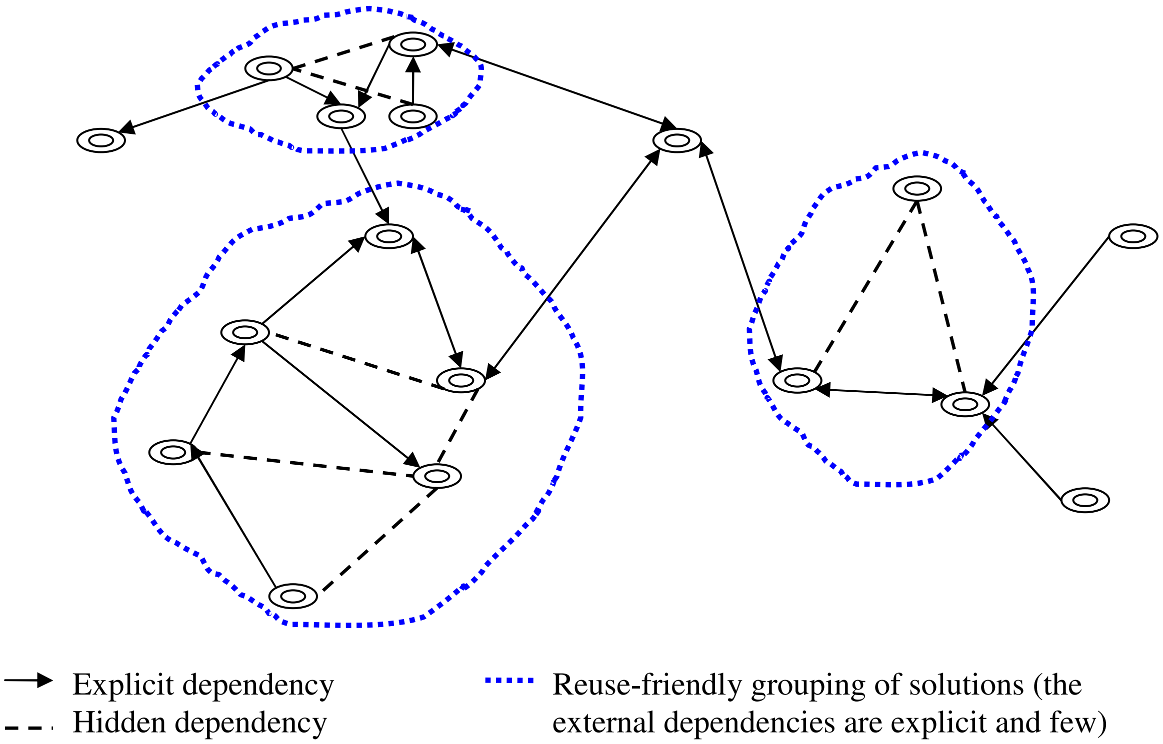
3.4.3 Reusable design solutions

Considered types of design solutions
This study focuses on the reuse of 'finalised' design solutions (as opposed to conceptual ones), because the study intends to analyse, among other aspects, the balance between product-specific optimisations and multi-product considerations. Such design solutions include:

- Standard components, reused as-is as black boxes
- Configurable components, partially reused (e.g. scaling) as white boxes
- Reusable fragments of components, that can be integrated into new components
- Structural design solutions (arrangement of components, internal interfaces, etc)
- Other design solutions: choice of material, manufacturing process type, user interface, etc

However, large parts of the reasoning may actually also apply to more abstract design knowledge.

Encapsulation of a reusable design solution
In order for a solution or a group of interdependent 'sub-solutions' to be regarded as a *reusable element*, it must have unambiguous, well-defined functionality and interactions with other solutions in the product. With these criteria, a part that has *hidden* dependencies to other parts is not convenient to reuse. Design solutions that are highly interdependent are practically inseparable and will need to be reused in group (or not at all). Therefore, a reusable element contains the notion of external independency, just as modules [Baldwin & Clark 2000] as is discussed later. Encapsulating a reusable design solution, that is, deciding the smallest independently reusable element of a group of complementing solutions may in some cases be trivial, but in other cases a considerable challenge, such as when dealing with complex reusable subsystems.



Harlou [2006] proposes the term ‘standard design’ to denote a reusable physical design or design principle which, in addition to being encapsulated into a self-contained functional unit as above described, must comply with three requisites:

- decision of reuse: standard designs are predetermined for future reuse
- documentation: standard designs are documented *for reuse*
- responsibility: standard designs have ‘owners’ who are responsible for correct implementation and changes

Internal interfaces: dependencies to other solutions

The internal interfaces are those between the design solution and other design solutions needed to make the product function as desired. The complexity of a component’s interfaces is an important factor determining its reusability. Some design solutions have very simple interfaces and pose few constraints on the product structure, like nuts and bolts. Other solutions have complex interfaces and behaviour that pose considerable constraints on the product structure. Martin and Ishii [2002] use the concept of ‘coupling’ between components to denote the probability that a change in one component will imply a change in the other. Martin and Ishii make a distinction according to the direction of the dependency, into *supplier* and *receiver*. ‘Supplier’ components are relatively insensitive to other components specifications, but their specifications have impact on ‘receiver’ components, and vice versa.

3.4.4 How can solutions be reused?

Reusing ‘form’ vs. ‘content’

We can differentiate between a) reusing structural solutions (‘form’), i.e. redesigning the components but preserving the interfaces, and b) reusing components (‘content’), i.e. rearranging structures. Of course, the categorisation into reuse of structure vs. components is a theoretical construct, and thus dependent on the viewpoint and level of abstraction.

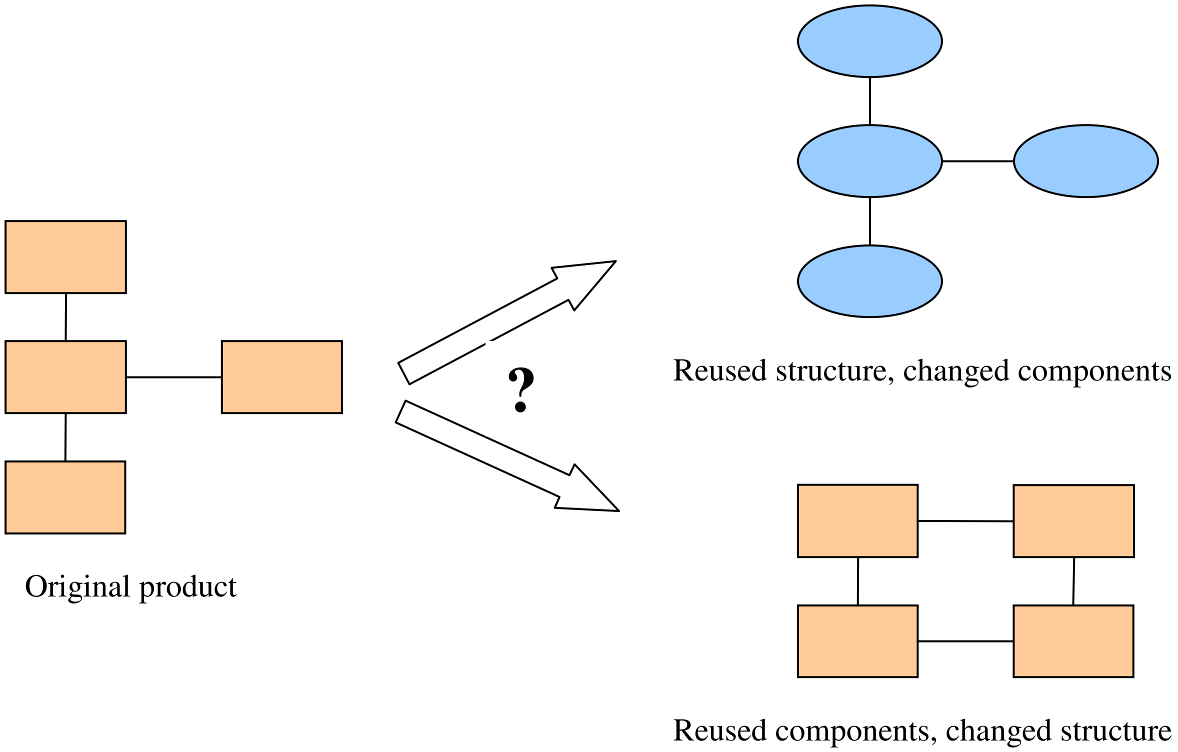


Figure 25: Reusing structure vs. components

In some cases there will be a choice between where to locate technical complexity. For example, a simpler product structure may need more ‘intelligent’ components, and vice versa. The manner in which this is solved can affect the possibility to reuse structural elements versus components. It may be that simpler structural solutions are easier to reuse (provided they are flexible enough), but force components to carry more complexity thus making them less suitable for reuse, and vice versa.

An example of reusing components in different product structures can be found in much IKEA furniture. In these, it is easy to find reused ‘low level’ standard components and design solutions like assembling interfaces, whereas the product structure can be very innovative and contribute to great spatial and generational variety.

Henderson and Clark [1990] discuss specifically the differences between changes in product structure vs. changes in content – which they refer to as ‘architecture’ and ‘components’, respectively. Henderson and Clark argue that organisations tend mirror the product architectures by means of organisational structures and information channels, which creates organisational inertia that hinders architectural innovation. They warn that innovation of the product architecture can have deep consequences on the organisation of product development, production and distribution – thus risking being counterproductive for business if not carefully planned and executed. Innovation of components, on the other hand, will tend to have more limited operational consequences.

Reusing structural solutions

Reusable structural solutions may be internal/external interfaces, component arrangement and allocation of functions to components. Structural solutions can be classified into [Sosa et al 2003]:

- modular systems, whose *"design interfaces with other systems are clustered among physically adjacent systems"* or
- integrative systems *"whose interfaces are physically distributed or functionally integrative across all or most other systems"*

Many examples of reuse of structural solutions can be found in the computer industry, where the standardisation of microelectronics architectures, including processor and bus interfaces, has allowed for great innovation 'inside' components.

Meyer argues that, in line with modularisation, standardisation efforts should begin with the structure (subsystem interfaces), and then go onto the components themselves.

> *"An effective approach towards standardization is one that is highly selective, carefully choosing elements that should be standardized. First and foremost are the subsystem interfaces (and user interfaces fore systems and software). Once robust interfaces are either designed or obtained from the industry at large, and then fixed into place, degrees of freedom emerge for developers to improve particular subsystems or to add entirely new ones."* [Meyer & Lehnerd 1997 p121].

This is actually what was done with computer electronics, an example of successful modularisation to face technological uncertainty:

> *"For an industry like computers, in which technological uncertainty is high and the test way to proceed is often unknown, the more experiments and the more flexibility each designer has to develop and test the experimental modules, the faster the industry is able to arrive at improved versions."* [Baldwin and Clark 1997, p85]

Morgan and Liker provide an example of how structural design solutions are reused at Toyota, where an aspect that is considered critical is

> *"the application of common architecture through detailed design standards and specifications, which the body engineer can draw from a database of best-body sections for each vehicle type. Body engineers can expand, shrink, or otherwise modify these structural best practices while the database simultaneously maintains critical geometric relationships to preserve product performance and manufacturability"* [Morgan and Liker 2006 p.54].

Another type of 'structural' design solutions are the interfaces to different supply chain phases, such as shapes that allow access to manufacturing tools, holes for attaching to testing rigs, lifting handles, etc. Achieving commonality with respect with this kind of design solutions is usually vital for an efficient production. Toyota, to take one example, applies strict standardisation of the critical geometrical aspects.

> *"Certain shapes, forms, and holes in exterior-stamped sheet metal and subassemblies necessary for efficient manufacturing (or successful vehicle crash performance) are identified and standardised across certain specific vehicle models or generations of the same vehicle."* [Morgan and Liker 2006 p.43]

Many authors have studied the impact of reuse of interfaces on the product development efficiency. Chen and Liu [2005] agree that standard internal interfaces may restrict changes but may bring several benefits:

- the components being designed acquire an explicit (and hopefully proven) interaction with the rest of the product, thus making their behaviour more predictable

- facilitate the coherent break down of the design work into smaller design packages that different teams can work with autonomously
- facilitate maintenance and updates

Chen and Liu also propose a set of possible interface strategies for modular products and discuss the technological and organizational requirements for each strategy. Sundgren [1999] specifically addresses interface management for product families with the purpose of improving the balance of time to market with the utilization of design commonality.

Reusing as-is

For the reuser, the advantages of reusing without modifications are:

- It is only needed to know how to *reuse* the solution, not how to design it.
- It may be possible to buy off-the-shelf.
- The solution will probably be less technically risky as it has been proved before.

A frequent downside of reusing as-is is that the solution may be one-size-fits-all meaning it is performance-wise suboptimal or more costly to produce, which we call over-design. The downside may also be more hidden. The reused component may seem ‘optimal’, but may force the adaptation of *other* design solutions in the product to make them compatible, as sort of technical integration cost of reuse.

Reusing partially

This is the case of scalable designs (meant to be modified), and designs that are to be improved. The main benefit for the reuser is being able to draw from the design knowledge previously developed and still be able to optimise the solution for the current product. The drawbacks may lie in the potential need to (re)learn about the details of the design solution and a higher technical risk than reusing as-is (although of course often lower risk than designing from scratch). Well prepared scalable components will of course be rather trivial to modify, in comparison with highly coupled designs where the ripple-effects of changes are difficult to foresee.

3.4.5 How to design by reuse

Consider a product under design that is to satisfy a given set of product requirements, and there is theoretical room for choosing what to reuse. In some cases, an available previous solution will turn out to be optimal for the given design problem, and then obviously it should be reused. Very often though, an available previous solution will turn out to be, from one or more viewpoints, suboptimal. These viewpoints can be time-to-market, product performance, supply chain costs, product strategy, etc.

If the previous solution is reused as-is, it means the sub-optimal aspect (over-design) is deemed acceptable compared to the benefit of reusing. If there are possibilities to modify the previous solution, the redesign and testing effort and risk of technical side effects must be considered. So the question is: how suboptimal can the previous solution be, before it is better to design a new solution? The answer is of course dependent on how the company *weights* different costs and benefits, that is, the company’s *reason* or *driver* to reuse.

If the focus is on saving development effort (costs and time), the following could be examples of situations when it is recommendable to reuse:

- scarce expertise: previous solutions where considerable expertise has been invested that is costly (or unnecessary) to replicate. To avoid the need of the scarce expertise, it will often be the case that the solution has to be reused *as-is*. This can of course be both

complex and simple components that the company does not have interest of spending resources to learn about and innovate.
- abundant expertise: previous solutions that are so well known to the developers that their reuse is spontaneous and efficient. Often the expertise will provide possibility to *modify* (improve or customise) the solutions to the benefit of the company.

In the case of 'face-lifts' or product configuration projects, of course the technical challenges and risks are not as important as upon original development, and the reuse choices (most elements will be reused) will normally be dictated by explicit product requirements.

If the focus is to decrease the supply chain costs, it is important to identify variety cost drivers and trace these to design parameters. At the same time, the design parameters that are 'cheap' to change (between limits) to create the desired product variety should be identified. For example, one manufacturing tool that needs to be reused may impose a length constraint on the component to be designed, otherwise allowing design freedom on other parameters. Design for X (DFX) where X stands for different cost optimisation areas as Manufacturing, Assembly, etc, is a set of tools that can assist engineers in identifying acceptable ranges for design parameters that drive costs in different stages of the supply chain.

Hence, high product value at low cost may often be created by *modifying* and reusing previous designs in the right way. However, sometimes a significant cost driver is the unique part count. Then, there is a need to reused previous components *as-is*.

3.4.6 How to design for reuse

In general, designing for reuse, as opposed to designing for one-off application, requires more knowledge and effort. When designing for reuse, it is often recommendable to invest in higher quality, considering the possibility of reuse in several products and the following higher volumes. Fisher has observed that higher quality often (but not always) is the case when components are designed to be shared, because of "*the learning and quality improvement associated with increased volume, and because increased production volume may justify higher levels of investment in component development and refinement.*" [Fisher et al 1999, p.299] Designing for reuse usually also demands a significantly higher *architectural* knowledge (understanding of the solution's role in the life cycle of the product and business context).

If the solution is to have more applications than one product, it will probably be subject to varying requirements [Martin & Ishii 2002]. Such drivers for change may be
- performance requirements on component (customer driven change)
- need for improvement (enabling technological evolution)
- production system changes
- need for cost reductions

The design solutions that make up a product can be classified in a range according to how *known* it is that they will be reused, from
- one-off solutions (designed without consideration for reuse), to
- solutions known to be reused in future

In between these extremes there are design solutions that are potentially reusable, but whose future possible reuse has not been decided. For potentially reusable solutions it is often not obvious *how much* development effort and technical compromise is justified. This study is above all relevant for this kind of design solutions.

Product structuring when designing for reuse

> *"Product architectures are essential to separate the stable and variable parts of design. The stable aspects create a framework within which a variety of products can be developed."* [Andreasen et al 2001 p.43]

To make interdependent design solutions reusable, either their dependencies should be reduced and made explicit, to allow their individual reuse, or they should be modularised, i.e. their external interface and function should be made explicit, so that they can be reusable in group. Designing for reuse often requires finding robust solutions that can be fit into 'unexpected' applications. Martin and Ishii [2002] propose a method called Design For Variety to design product architectures that deals with change over time. The method includes identifying external factors that are likely to lead to design change, and isolate pertinent components, so that the rest of the components and product structure can be reused despite the expected changes. Design For Variety uses the concept of 'specification flows', together with two indexes: a generational variety index (*"the amount of redesign effort required for future designs of the product"*), and a coupling index (*"the coupling among the product components"*). These two indexes are used by the design team to *"develop a decoupled architecture that requires less design effort for follow-on products"*. Design For Variety appears most applicable in cases where the probable changes are fairly predictable and limited in scope – i.e. their propagation can be isolated by choice of product structure and tolerances.

Modularisation

Modularisation methods aim at designing modules with well-defined interfaces and clear allocation of functionality, which often makes them more suitable for reuse [Baldwin & Clark 2000]. Modularisation is based on the *"decomposition of a product into building blocks (modules) with specified interfaces, driven by company-specific strategies"* [Ericsson and Erixon 1999, p.19]. The modularisation of products is driven by a number of 'module drivers' related to business needs [Ericsson and Erixon 1999]. The module driver that we consider here is design reuse, or carryover as Ericsson calls it.

Baldwin and Clark describe the method of modularising as separating the design parameters into visible design rules which affect subsequent design decisions and should be defined early in the design process, and hidden design parameters, which are

> *"decisions that do not affect the design beyond the local module. Hidden elements can be chosen late and changed often and do not have to be communicated to anyone beyond the module design team."* [Baldwin & Clark 1997 p.86].

Visible design rules are of three types:

- the architecture, specifying the modules that will compose the product
- the interfaces that state how the modules are to interact, and
- testing specification for the modules

Modularisation can be used to help rationalise the product development organisation, by allowing the separation of module designing into more autonomous teams. Modularisation can therefore be a powerful organisational tool, allowing for clearer boundaries and communication channels between company functions and development teams. This in turn facilitates the concurrent execution of different development tasks [Sanchez and Mahoney 1996]. However, modularising the product and the organisation can in some cases be very challenging, as noted by Baldwin and Clark:

> *"If modularity bring so many advantages, why aren't all products (and processes) fully modular? It turns out that modular systems are much more difficult to design than comparable interconnected systems. The designers of modular systems must know a great deal about the inner workings of the overall product or process in order to develop the visible design rules necessary to make the modules function as a whole. They have to specify those rules in advance. And while designs at the modular level are proceeding independently, it may seem that all is going well; problems with incomplete or imperfect modularisation tend to appear only when the modules come together and work poorly as an integrated whole"* [Baldwin & Clark 1997 p.86].

However, when 'just' modularising specifically *for reuse* (encapsulating the reusable design solution), then the aim is not to organise the development work. Then, there is not an *organisational* need to define the interfaces early in the design process, and interface specifications could be left floating and alternatives explored until more knowledge allowed for confident decisions.

The duality of potential effects of modularisation on product development, i.e. reuse and design work coordination, is mirrored by a survey by Ishii and Yang [2003], where the respondents anticipated, based on professional experience, two main benefits of modularisation with respect to product development, namely

> *"concurrent development and design solution re-use. If a product consists of modules with well-defined interfaces, each module may be developed in parallel as opposed to serially. This would obviously save a significant amount of development time. When a company develops generational variety, only those modules of concern must be worked on."* [Ishii and Yang 2003]

3.4.7 When future requirements are known: product variety optimisation

When the needed variety is known, designers can explicitly solve the variety optimisation problem, that is, co-optimise variety with commonality. This is the case when sharing solutions across simultaneously existing products, especially product families where the products complement each other in the market. The question of *what to share* across concurrent products is what we could call the 'design sharing problem' or *product variety optimisation*, which could be formulated as: finding the optimum combination of shared and product-specific design solutions that minimises the total costs of a product family with given functionality requirements. Another formulation of the same problem is: *"given a product portfolio, how many versions of each type of component should exist, and what subset of products should use each component design?"* [Fisher et al 1999 p.298]

Arguably, the main challenge in the design sharing problem is not conceptual, but comes among other factors from the practical difficulties of quantifying the true long-term costs and customer perceptions of the alternatives.

Several authors have contributed to solve this 'sharing problem', e.g. by choosing size ranges for shared components [Fisher et al 1999][Pahl & Beitz 1996], or by choosing a product structure that enables a better balance between commonality and variety in across the product family members [e.g. Ulrich & Eppinger 2003, D'Souza & Simpson 2003]. Ramdas et al [2003] present a model of component systems sharing in assembled product lines, and suggesting optimisation approaches to know which components should be shared. Messac et al propose a *"product family* *penalty function"* in an attempt to visibilise the cost of variety of for different design parameters. The aim is to support the selection of common and scaling parameters in scalable platform-based products, using the 'physical programming paradigm' to identify the determining physical parameters [Messac 2002].

Also more 'mathematical' contributions have been made. Fujita [2006] proposes a solving multi-criteria optimisation method that has the objective of minimising the total cost or maximising the profit. Fujita argues that a simple trade-off between performance and direct cost typically applicable to single-product optimisation, can not apply to multi-product optimisation. This is because different products will have different trade-off patterns that

> *"include various issues over the production and utilization of all units of different products. Under this reason, total costs or total profit through all designing, manufacturing and utilizing multiple products can be the objective rather than simple performance or direct cost."* [Fujita 2006 p.192]

One requisite for solving this problem with mathematical tools is that the design problem must be possible to translate into formal style. Design variety optimisation is applicable when a range of expected product functionalities, with associated costs and constraints, can be predicted or predetermined with considerable accuracy, because *"interpretation of customer's needs is indispensable as the prerequisites for the optimization"* [Fujita 2006 p.191]. When the uncertainty becomes too large, design sharing optimisation tools may prove inappropriate.

The 'design sharing problem' is concerned with a set of *simultaneous* products. This has two implications that are important to highlight here. The first is that the company context, notably technological and market knowledge, can be assumed constant during the development of the considered products. The second assumption is that there is negligible cost for transferring design solutions from one product to another. Designs can often be tested in several of the products under development that are to share them, and real-time feedback loops can facilitate adjustments.

3.4.8 When future reuse requirements are uncertain

When future applications in which the design solution may serve are unknown, the designing for reuse is not an optimisation exercise, because there are no defined optimisation criteria. Rather, the designing can aim at making the solution as flexible and robust as possible, without excessively compromising the solution's performance in the current application. Berglund and Claesson [2005] propose the notion of a *design bandwidth* to denote the solution space limited by the minimum required variety (flexibility) and the minimum required commonality. They argue that this solution space should not be viewed as one-dimensional – i.e. a trade-off between commonality and variety – but as a multidimensional space where clever designs can satisfy *both* high commonality *and* high variety objectives. In this context, *robustness* is the technical property that can be designed into a solution as a response to uncertainty in future demands

> *"Paradoxically, reuse-intensive development is best accomplished by focusing more on how to change software effectively than how on to keep it from changing"* [Barnes & Bollinger 1991].

As with software, designing mechanical products for robustness is mostly about identifying elements of the design that unnecessarily pose restrictions on the use of the solution, and either eliminating these elements through rationalisation of the solution, or externalising them, so these elements can be adapted (*"ported"*) to the application without changing the reused solution. As a rule of thumb, designing for robustness should not make the solution more complex to handle several envisioned applications, but simpler to be easily adapted/completed to "unimaginable" applications. If the future requirements are unknown and the probability of meeting the future requirements are small, any significant extra costs in designing effort, compromised performance in the current application and/or increased complexity of design knowledge transfer, will make reuse uneconomic.

In some situations, design teams designing for future reuse will have to make a choice between increasing the complexity of a reusable solution to meet a number of future expected applications or reducing the complexity to make it easier to modify the reusable solution. This has for instance been observed in a case study concerning reuse in one electronic product development project by Ahonen and Nurmi. The study concludes that

> "*the architecture of a reusable block should be designed to cost-effectively enable every imaginable solution. However, the need for modification cannot be avoided for all possible future applications. Hence, maybe the most crucial discipline in producing reusable designs is to facilitate modification. This is best achieved through consistent, self-explanatory style of description.*" [Ahonen & Nurmi 2004]

It should however be stressed that often there is no conflict – because the *consideration* of several imaginable applications during design of a reusable solution may help arrive at a *simpler*, more elegant and adaptable solution.

Designing for robustness often involves providing 'generous' tolerances to the design solution. By designing tolerances to accommodate future change and customisation, the chances improve that the design may be reusable [Eckert 2004]. Jiang and Allada propose a methodology for designing a modular product family to facilitate adaptation to future changes, with the aim to improve the robustness against changes in customer requirements. Their methodology implies modelling the design process as a *control system*, where customer requirements are modelled as signal factors, future changes of customer requirements as noise factors, and a quality dimension is used to evaluate the product family [Jiang & Allada 2001].

3.4.9 Conclusion

What do we call a 'design solution'?
A product's design is constituted by the sum of all its design solutions. A design solution is a decision made in response to a design problem. The term can be used flexibly to denote differing levels of abstraction and detail.

How can solutions be reused?

- Solutions can be reused as-is (black-box) or partially (white-box).
- Both components, partial components and structural design solutions (interfaces, physical arrangement and allocation of functions to components) can be reused.
- A reusable element must be 'encapsulated' (i.e. have defined interactions) to be reusable.

What are the criteria to know if a past solution is reusable?
If there is fit between current product requirements (functional, performance and supply-chain requirements) and the past solution, the decision to reuse is trivial. But if the past solution is in any way suboptimal, the alternatives are:

- Modify it (optimise for current product), or accept the technical sub-optimality cost
- If to reuse it as-is:
    - how much must other aspects of the product be compromised?

- If to modify it:
    - are chain effects foreseeable? (transparency of design)

How to design for reuse?
During *conceptual* design: reuse-aware product structuring can yield naturally reusable design solutions that can be useful in many products. This should be done by mapping functional requirements to design solutions so that areas of low change likeliness and low dependency to other solutions are grouped into reusable elements.

During *embodiment* design:

- If to design a black-box reusable solution: make it robust, dependable, easy to use, etc. A black-box reusable solution can have higher internal (hidden) complexity for the sake of better external performance and simplicity.
- If to design white-box reusable solution: make it transparent, pedagogical, scalable, that is, prepare it for changes, make it customisable and document design intent to facilitate understanding.

3.5 Knowledge management view: How to transfer design knowledge for reuse?

This section analyses which design knowledge is needed for design reuse, and how this knowledge can be supported by transfer of design information. In addition, available tools to support such information flow are discussed.

3.5.1 Questions

1) What is the role of design knowledge and information in design reuse?
2) What design information is needed for successful design by reuse?
3) How should design information for reuse be represented for efficient search and retrieval?

3.5.2 Introduction: organisational learning

For clarity, let us first distinguish the terms knowledge, information and data. Knowledge is a cognitive asset of living beings (there is no knowledge in paper or computers). Data is a representation of something, which need not have an intended public and may or may not make sense to the observer. Information is data that deliberately has been put together for a public, and that has a meaning to the observer. So the same piece of data can be information for an observer that can understand it, and just data for another observer that can not make sense of it.

The challenge of inter-project learning
Effective organizational learning, however obvious its benefits might be, appears for different reasons to be difficult to achieve. This is for example indicated by a study of 120 technology-based companies which found that only 50% of the interviewed R&D directors claimed to be practicing the "launch-learn-launch-mode" i.e. systematically trying to learn from product development projects and applying experiences in subsequent ones [Gupta & Wilemon 1996]. Similarly, Huang and Newell claim that *"organizations often fail to generate insightful lessons from their implementation experience"* [Huang & Newell 2003]

The causes for limitations in organisational learning are various. Gann and Salter mention some possible organisational reasons:

> *"The limits of knowledge management techniques are not only driven by the project-based nature of activities, they also arise from high turnover, reluctance on the part of engineers to recycle designs and an incentive system within the profession, which rewards novelty rather than standardization."* [Gann & Salter 2000, p.969]

For many organisations, the main problem may be simply that they are too stressed fire-fighting:

> *"An organization cannot learn effectively simply by concentrating on fixing problems with quick solutions. Rather, it is vital to promote generative learning by constantly evaluating the way in which solutions were created"* [Huang & Newell 2003 , p.174].

Finally, another possible problem is related to cognitive factors, because

> *"learning in the firm tends to be local. Interpretation of experience is difficult, as lessons must be drawn from a relatively small number of observations in a complex and changing environment. This makes it laborious to identify causality and draw correct inferences. Organizations and organizational members exhibit systematic biases in interpretation, since they concentrate overwhelmingly on recent and salient events. Also, they may be insensitive to sample size, attribute too much importance to intentionality, and may use simple and linear algorithms."* [Prencipe & Tell 2001, p.1373]

For example, sometimes wrong conclusions are drawn from past decisions by taking them out of context or assuming they were more considered than they actually were.

Organisational approaches to enhance inter-project learning

One approach to enhance the transfer of knowledge across product development efforts is based on continuous project management. Continuous project management, according to Starr,

> *"takes the form of an on-going effort to manage a stream of multiple new products. Teams are permanently assigned to project development activities. Often the goals are targeted incremental improvements, but they also can be directed toward significant 'breakthrough' versions to radically alter a previous product design. [...] The system has a memory for ideas with potential utility that had to be shelved the first time around because they did not fit the schedule and/or match the prior goals. The capacity to reexamine and learn from prior project steps that were taken is very important."* [Starr 1990, p.89]

Part of every project assignment is to learn how to improve the next project, and projects need to be supported by organisational planning that ensures that feedback loops allow inputs to projects are effectively transferred from the projects were they are originated.

Another approach to effectively exploit organisational learning is the usage of central knowledge bank groups. Gann and Salter claim that *"modern forms of apprenticeship and peer group and team-based learning appear to offer important mechanisms for overcoming [learning] discontinuities."* Some companies have tried to solve this with *"support groups [which] act as repositories of knowledge and information about firm-wide processes."* These groups gather experiences from past projects and try to transmit the knowledge to new projects. However, *"because project-teams are often self-contained, they may draw little from central services such as technical and R&D support."* [Gann & Salter 2000, p.968]

For reading about an evolving comprehensive model of learning in design, we refer to Sim and Duffy [2004]. In the following sections, we explore the main issues of knowledge transfer for design reuse, and available tools to support it.

3.5.3 Design knowledge and its representation

Designing is a form of problem solving centred on understanding the design problem. The process of designing yields design knowledge to the designers. The knowledge that the designers are conscious of and can articulate is called explicit knowledge, while the knowledge that the designers possess but are unaware of is called tacit knowledge. One part of the explicit design knowledge is represented in production specifications and user instructions. Other design knowledge might be documented in design notes, meeting protocols, etc. However, the largest share of the design knowledge is never documented. Of course, it is not possible or even desirable to represent it all, because the design knowledge is so vast and complex and contains much 'noise' in the form of fragmentary and even contradictory pieces of knowledge.

To the designer, a design solution may mean an understanding of a design problem an idea how to solve it and related tacit knowledge. To the company, besides being organisational knowledge, design solutions are information embodied in different media:

- documentation
    - requirements stating the solved design problem
    - specifications for production
    - virtual models
    - user manuals
    - documentation for maintenance and reuse
    - test specifications and test results
    - project protocols and notes
- implementations such as components and prototypes
- supply chain infrastructure
    - production tools
    - test tools

3.5.4 Design language

One important way of increasing the effectiveness of the knowledge acquisition by the reusers, is to use a design language that is precise, efficient and unambiguous. This can for example be a formal modelling language complemented by working language [Andreasen 2001][Smith & Duffy 2001].

Genetic design model system and PFMP

There is a rich history of modelling languages for the designs themselves. Different engineering fields have different well-used modelling, most often adapted for a rather low level of abstraction. The Theory of Technical Systems [Hubka & Eder 1988], provide an excellent basis for conceptual modelling of mechanical systems, which other scholars have developed further. For example, [Mortensen 1999] proposes a "genetic design model system" based on constitutive (what is part of what), and behavioural (what does what) models. Interestingly, Mortensen's contribution is very useful to model the role of a single solution in the entire part assortment of a company, which is potentially very interesting for a reuser.

The genetic model has been used as a basis for a model to represent entire product assortments in a so-called Product Family Master Plan, or PFMP [Harlou 2006]. The PFMP provides an overview of the product assortment by showing its structure and variety from three viewpoints: customer view, engineering view and part view. The modelling uses a formalism based on 'classes', 'attributes' and 'constraints' in two types of structures: 'part_of' and 'kind_of'. This formalism allows for a graphical and clear modelling of the most important relationships between the design solutions that constitute the product portfolio. This overview is very useful to identify unnecessary variety, and discuss the potential impact of changes in the product assortment.

Configurable components

Claesson [2006] proposes the concept of *configurable component* to support the development of configurable products such as platform derivatives. The configurable component is a modelling formalism suitable for incorporation into design tools to support distributed design coordination and modelling of evolving (not finished) parameterised designs. Claesson provides the following description of the concept:

> '*A configurable component is an element that has been defined in order to allow for definition of different variants of the configuration of a system. A configuration of a system is a specific variant of the system that has been achieved by selecting values for the variant parameters made available in the definition of the component. The set of variant parameters that are available for requesting a configuration of the component are3 collectively referred to as the variant parameter interface of the component."* [Claesson 2006 p.98]

Design ontology and other tools

There are also various attempts to provide a design language based on a design *ontology*, i.e. a formalisation of design concepts and their interrelationships. For example, Nanda et al propose a Product Family Ontology Development Methodology that seems very promising for reuse-oriented design documentation [Nanda et al 2006]. Li et al [2005] have analysed the needs of industry in this area and the challenges for academia, and conclude that

> *"We believe that ontology representation is the crux of knowledge systematization by providing the theory of the content and the mechanism of inference. It structures the domain knowledge based upon different perspectives. An ontology model provides guidelines for capturing the target domain and indices for knowledge retrieval."* [Li et al 2005, p.12]

The question of how design information should be structured and indexed is also related to the appropriateness of ontology (i.e. predefined categorizations of information), versus 'folksonomy' in different cases. It appears that ontology may be suitable for expert systems that evolve in a controlled manner.

Further interesting reading can be found in [Crossland 2003], who proposes an object-oriented modelling framework for representing uncertainty in early variant design, and [Dalton 2005] who presents design modelling frameworks to capable of representing traceability information such as design rationale to support design reuse.

3.5.5 The information needed when reusing

To *reuse* a solution, designers need to gain particular knowledge that is different from the knowledge needed to *design* the solution. Companies normally do not have the know-how to design all the solutions they reuse. Some solutions may have been bought from suppliers who did not transfer the full related know-how with their delivery. Other solutions may have been designed in-house, but the associated know-how has been ‘forgotten’, leaving what we could call ‘legacy’ solutions.

To gain the knowledge necessary for reuse efficiently, the reusers need *information* about the design solution considered for reuse. Reusers can learn the needed knowledge by other means than reading design documentation, for instance by reverse engineering of physical implementations of the design solution, experimenting, etc. Therefore, the issue of design information transfer is to a large degree a *pedagogical* one: How can the project that creates a design solution help the subsequent reusers to *learn* how to reuse the solution?

The type of needed knowledge depends on whether the reusers are to modify the solution or if they are to reuse it ‘as is’. When the reusers must redesign previous solutions (white-box reuse), they need deep design knowledge, sometimes comparable to the knowledge of the original designers, depending on the nature of the modification. When reusing solutions as-is (black-box reuse), the reusers need ‘just’ know when and how to reuse the. To this end, they need instructions about the solution’s range of performance, restrictions and interfaces, but theoretically they do not need information about the inner workings of the solution. However, if the available documentation about how to reuse the solution is deficient and/or insufficient, the potential reusers will need to investigate themselves, in which case information about the inner workings of the solution and design rationale can be useful.

Baldwin and Clark state that

> *“Designers’ knowledge of prior designs includes things such as*
> - *copies of older artefacts in the class or category;*
> - *detailed information on design parameter choices for those artefacts;*
> - *known problems or shortcomings of those artefacts;*
> - *results of various functional tests (e.g. megahertz ratings or access times);*
> - *data on user acceptance and overall market value (survey evaluations, product reviews, estimates of units sold, and prices).”*
>
> *“Less easy to pin down, but also available to designers, are the remembered task structures, which were used to create the prior designs...”* [Baldwin and Clark 2000, p.125].

To the previous list, we could add other pieces of information that might be useful to the reuser:
- relationship to complementary and alternative solutions, comparison to similar solutions
- design intent
- performance ranges and restrictions, not only in the original product
- reuse advices and references to further sources of information (like the names of the original designers). For example, solutions could be tagged with reuse-guidance, e.g. ‘reuse recommended’, ‘obsolete’, ‘temporary adaptation/experiment’, etc
- description of the designing steps used to arrive at the solution, to evaluate quality and to replicate problem solving procedure

3.5.6 Documenting for design reuse

How to document designs so that future designers can and want to reuse them?

Design rules

Design solutions prepared for reuse should include reuse instructions that are a form of 'design rules' [Baldwin & Clark 2000]. Design rules can be seen as *contracts* that specify what a solutions shall do and how *other* solutions interacting with it are to be designed to be in harmony. Therefore, design rules define the role of the solution in a product architecture. For example, in platform based development, those designing the product platform architecture impose a set of design rules specifying how the product variants are to be designed.

Design rationale

Design rationale is one of the most important pieces of information that is specifically useful for design reuse. Unfortunately, the documentation of design rationale is resource-demanding, and because it is often considered a 'co-product' by development projects, it often gets down-prioritised when resources are scarce [Busby 1999]. Ball et al argue that

> *"design reuse is plagued with difficulties, including those associated with the indexing, retrieval, understanding and modification of prior design knowledge"*, and propose that companies should learn from *"design-rationale research concerning how best to represent and retrieve design information"*, in order to maximise *"the benefits of rationale capture and information retrieval whilst minimising the costs to the designer that might arise from disruption to natural design work"* [Ball 2001].

One of several other related research contributions is the one of Kim et al [2005], who propose a framework for design rationale *retrieval* to support design reuse, based on a semantic-based retrieval method that 'understands', organises and extracts information in a way that 'mimics human thought'. This way, they claim, less requirement on formalising the design information and query terms are posed on the original solution designer and the reuser.

Prioritising objects of documentation

One challenge is to identify which parts of the design knowledge need to be 'captured'. This should balance the effort needed to document and maintain information for reuse with the benefits to the future reusers. Maintaining available design information up-to-date is crucial, to ensure that its user *trust* it. Otherwise, information users may need to devote considerable effort verifying it, thus ruining the efficiency of the knowledge transfer, or worse, misleading the information user to erroneously trust an incorrect piece of information. To provide further nuances to our discussion, we may classify design knowledge according to how it evolves, from 'hot' to 'cold' knowledge.

- Hot knowledge is dealt with every day, and therefore is difficult to document (it evolves continually, and it is so 'obvious' that nobody knows how to explain it), documentation could be redundant (everybody is anyway 'in it').
- Cold knowledge could for example be knowledge that was gained once but is not needed anymore, or for a long time (so long that it is acceptable to gain it again from scratch when/if that time comes again).
- Key knowledge that is scarce in the company: this should have the highest priority to be documented: you use it so seldom that you have time to forget about it in between, but you need it regularly, so it is worth the job of documenting it.

3.5.7 Disseminating design documentation

Which channels should be used for the communication of knowledge from earlier projects to new ones? Often, because of lack of more formalised channels, project members resort to ad hoc search for information, as expressed by an interviewed product development manager:

> *"knowledge codified in project documents does not get diffused properly. [... It] gets transferred more easily via informal channels, however 'if you don't know the person concerned, you will not go and talk to him'."* [Prencipe & Tell 2001, p.1386].

Malik [2002] proposes a 'broadcasting model' for intra-firm technology transfer, see figure below. Malik claims that the management of technology transfer often is poorly understood, and suggests that the 'broadcasting model' can be used to improve understanding of helping and inhibiting factors.

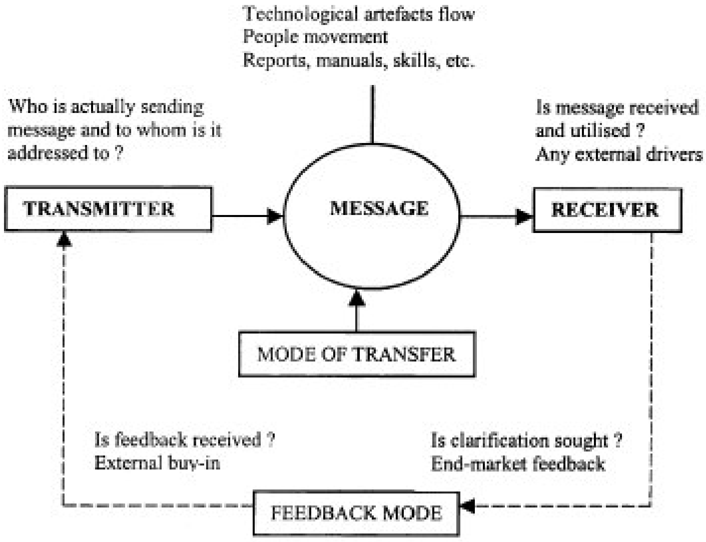


Figure 26: A broadcasting model for intra-firm technology transfer [Malik 2002]

The broadcasting model above can be used to structure our reasoning about reuse information transfer. We have discussed the role of the transmitter (the designer of the reusable solution) and the contents of the message (information needed to reuse a solution efficiently). Unfortunately, there is a risk that these get analysed in isolation. As the model suggests, companies need to consider the receiver as an active part of the system, *interacting* with the transmitter and the message, if the process is to be managed effectively.

In order to avoid redundant and ambiguous information, design information should be stored in a centralised searchable repository and indexed in a reuse-friendly way. Software tools are today indispensable for achieving this efficiently. Tools like PDM/PLM and more recently Knowledge Based Engineering (KBE) have greatly facilitated for companies to store and structure product information efficiently, and can probably in most cases be used to carry reuse-specific information about designs. Interestingly, Weber et al [2003] propose a tool evolved from PDM/PLM which they call Property-Driven Development/Design (PDD). PDD is based on a clear distinction between product *characteristics* (structure, shape and material consistency of a product) and *properties* (behaviour). Weber et al claim that PDD can increase the transparency of development activities thus making them easier to control and speed up.

Busby [1998] has studied effective practices in design transfer, by means of extensive qualitative interviews. At the studied companies, Busby identified the following means of improving design transfer:

- associativity-improving
- criteria-broadening
- effort-reducing

- environment-influencing
- error-adverting
- motivation-addressing

Without going into detail about each of these approaches, we can note the diversity of issues that engineers at companies consider worth addressing to improve knowledge transfer.

One challenge is how future reuse planners should be informed when there exists a reusable solution they should consider. This is because if the repository of reusable solutions is large, it can be difficult for reuse planners to know what to search for. This issue is notably addressed by Case-Based-Reasoning (CBR), which is a theory about how humans solve problems that has been successfully used as basis for tools specifically developed for knowledge reuse. CBR-tools appear to be very useful to assist in the transfer of design solutions [Belecheanu 2003] [Lees et al 2000]. CBR is based on a process that includes the identification of potentially reusable solutions, a procedure for documenting relevant design and search information, and a structuring of the information for later retrieval. If successfully implemented and managed, such tools can make it so easy to search and retrieve past solutions, that the design reuse practices at the particular company can be radically improved.

3.5.8 Conclusion

What is the role of design knowledge and information in design reuse?
To reuse a design solution, the reusers need to know in which circumstances it is suitable to be reused and if the solution is to be modified, how to do so, and how to design other interacting solutions to function in harmony with the reused solution. To acquire this knowledge, it is very useful for the reusers to have design information prepared by the original designers. The original designers of the solution should therefore try to anticipate which knowledge the reusers will need, and facilitate it by documenting selected pieces.

What design information is needed for successful design by reuse?
The following pieces of information are useful to the reusers

- original requirements stating the design problem
- specifications of the design solution including design rationale
- used design procedure, discarded alternatives
- test results and experiences from the lifecycle of the solution
- reuse advices

How should design information for reuse be represented for efficient search and retrieval?
First, it is important that an efficient design language is used. The design information should be documented and stored in a way that makes it easy for the potential reuser to search for it, especially based on a design problem, but also following other search criteria (similar solutions, specific products, originating project, names of the engineers, etc).

3.6 Organisational view: Who should decide on design reuse issues?

This section analyses how organisational aspects influence design reuse practices. We will mainly focus on the implications of the project organisation paradigm and the reuse-related decision structure and incentive systems. The analysis intends to cover typical practices, perceived problems and improvement areas.

3.6.1 Questions

1. How can the choice of project-heavy vs. functional-heavy organisation affect reuse?
2. How can the decision structure affect reuse practices?

3.6.2 Introduction: types of development organisations and their impact on reuse

Product development is typically organised as a matrix organisation that balances the needs of projects and the needs of functional teams. We can discern two typical organisation forms:

- The project-heavy organisation, where cross-functional teams are put together and co-located to work exclusively for the project under a strong project leader.
- The functional-heavy organisation, where projects are more loosely composed of representatives from different functional expertise areas that may work for different projects concurrently and are based with their respective functional groups.

The choice of functional organization vs. project organization depends on various factors.

> *"Functional units in a typical mass productions firm [...] lack the flexibility and responsiveness necessary to cope with unusually complex, new or rapidly changing project requirements. [...] Increases in functional specialization and economies of scale and scope exert pressures to move a firm towards a functional organization. Increases in task uncertainty, diversity and changing external conditions exert pressures to move a firm towards a total project organization. A matrix organization is required if opposing forces are equally strong."* [Davies and Brady 2000, p.937]

However, the trend in many companies is towards project-based organisations. Naturally, this has impact on the design reuse practices. For example, if the same individuals participate in two consecutive projects developing similar products, design reuse will tend to happen spontaneously. By contrast, if the persons that originally design a solution are different from the potential reusers in a subsequent development effort, a more deliberate initiative will be needed to make reuse happen.

> *"The goal of [project-based] organizations is to foster responsiveness and speed by granting autonomy and control. The consequence of this autonomy may be more difficult coordination with other activities within the firm"*, thus inhibiting reuse. *"Functional organizations, in contrast, may be less responsive and nimble, but they may foster greater coordination among projects and therefore easier sharing of components."* [Fisher et al 1999 p.299]

The tensions between projects and other functions of the company

Project-specific interests can sometimes come into conflict with the company's longer term interests. The reason for this can be business dynamics and incentive systems. In the pursue of predictable, deterministic (though potentially irrelevant) results, project organisations have interest in freezing requirements at an early stage, and isolate the projects from possible changes and input from the company environment [Kreimer 1995]. Also, one should be aware that in some cases competition *between* projects may hinder reuse, if it

> *"pits one project against another in the contest for limited resources. ... It is not a process that encourages collaboration. Projects that should be sharing core technologies, market insights, and product platforms are placed in competition. Sharing of resources, capabilities, and business opportunities occurs only by convenience or chance."* [Meyer & Lehnerd 1997 p230]

Therefore, it is important to ensure that different stakeholders' voice on reuse-decisions is balanced. Too often, the immediate stakeholders of the product development project (customer, project team, sales department) and the more long-term stakeholders, (e.g. future project teams, owners, maintenance department) have conflicting interests but are not able to articulate these equally at the moment of decisions.

3.6.3 Decision making on design reuse issues

As mentioned before, the central reuse-related decisions about particular design solutions are

- the decision to consider reuse of a previous solution to a new design problem and
- the decision to prepare a design solution for future reuse

In this section we will mainly regard them together as 'reuse-related decisions', because in this section we are only studying the form of this decision making.

Decisions within product development can be classified into "*decisions in setting up a development project*" and "*product development decisions within a project*" [Krishnan & Ulrich 2001]. At a more detailed level and focusing on individual reuse choices, we can classify the reuse-related decision making into

- decisions made *before* the actual development begins; this includes requirements, constraints and guidelines to the project based on reuse needs known before the development starts, from long time before or from the front-end phase of the project
- decisions made *during* the development project; this includes both active explicit decisions and spontaneous 'passive' decisions made upon the identification of 'new' reuse opportunities

Active vs. passive decisions

Project team-members have limited capacity regarding the number of active decisions they can make during a project, so they must prioritise. In cases where the number of potential design choices is overwhelming, the decision makers do not have the resources to consider alternatives and explore the solution space for all these choices. Therefore, a number of design decisions must be taken passively, 'by default'. This means the designers must 'ignore' some reuse choices, either relying on existing solutions without thorough evaluation, or redesigning new solutions without checking if already existing solutions would be better alternatives. Decisions are sometimes made and accepted gradually, 'semi-actively'. Active decisions may or may not be made explicit, i.e. articulated and accessible for others.

Strategic vs. tactical decisions

Who should decide what is allowed to be variant-specific and what needs to be reusable? Some reuse decisions have considerable strategic impact, while other only have tactical impact. While the impact of some decisions may be apparent, often the strategic dimension of decisions may be hidden. At periods, every detail may seem to potentially have strategic impact, because the frameworks determining the solution space have not been defined. A defined product, production and organisation architecture may help isolate issues of strategic impact from those of tactic impact.

Therefore, one challenge is to identify the appropriate decision forum for each reuse decision. For example, local-impact design decisions may be most appropriate for designers to make (guided by overall reuse rules), because if top-managers make low-level decisions (by so called 'micro-management') they may be bypassing valuable expertise of designers while being distracted from more urgent strategic issues. Decisions with strategic impact should be promoted to ‘product boards’ or alike; decisions that concern many company functions should be made by cross-functional teams, and so on. Some decisions may require company-wide commitment to be implemented, for example because they imply organisational changes as in the case of the design of the supply chain.

Hauser [2001] proposes a “metrics thermostat”, in analogy to an adaptive control feedback mechanism, to draw management attention to the most important issues and help the design team make strategically aligned decisions. According to Hauser’s method, management changes the weights of ‘implicit rewards’ to balance different strategic considerations, thus providing incentives for team members to make strategically sound low-level decisions without direct management involvement.

Finally, we note that some decisions require *coordination* with other decisions to be successful. This is the case with 1) the decision to prepare a solution for reuse and 2) the subsequent decision whether to reuse the solution, because the first is pointless unless the second is at least considered.

3.6.4 Deciding future reuse

An important factor that shapes the design reuse approach is the timing of the decision to reuse a certain design solution:

- in advance (reuse of the solution is decided to be mandatory in the future product(s))
    - enables design sharing optimisation
    - simplifies coordination
- when needed (reuse of the solution is optional in future products)
    - leaves more flexibility – less risk

A company may benefit from deciding in advance to reuse (‘freezing’) if there is high confidence in the prediction of future needs. This confidence can come from the fact that the company operates in static/predictable environments, or that the company has the flexibility to shape its future through ‘normative forecasts’. Arguably, whenever there is confidence that a future reuse opportunity will arise, companies should capitalise on this with a formal decision to reuse. This is because:

- such an early decision provides certainty that favour organisational commitment to get the most out of the reuse
- engineering teams can optimise the designs to be reused and related assets accordingly
- when the moment to reuse arrives, no effort needs to be spent on finding a solution to use (generation of concepts, evaluation, coordination)

See also [Baldwin & Clark 2000] who put forward strong arguments explaining the requisites and potential benefits of design decisions taken in advance, which they call ‘design rules’.

However, in many cases the future design needs cannot be predicted with confidence enough to justify the risks of making an unfortunate decision, so it is more convenient to defer/postpone the reuse decision. This critical level of risk should be identified to differentiate potential reuse decisions. This postponement decision should ideally be made explicitly, in order to assure that the necessary options are left open for the future.

One limitation of this modelling of reuse decisions is that in reality often many decisions are taken gradually.

3.6.5 Reuse opportunities identified *before* development

The potential benefits of reuse may in some cases be known before product development starts. This may regard particular solutions, which then can be pin-pointed to ensure reuse. A general need to reuse can also regard product/technology areas or supply chain resources. Sometimes the need is to keep the number of new solutions or the changes in production tooling, etc, as low as possible. Then what has been identified is an aspect of internal variety to optimise (minimise) for.

Requirement management

The problem with one-product-at-a-time design processes is that the input to the project is captured only by functional requirements and constraints. So where is the formal incentive to decrease costs in future products by designing by and for reuse? Arguably, if there is an identified need to reuse a previous solution or prepare a solution for future reuse, this should be *required* along with the functional requirements and production constraints posed on a development project. Juuti & Lehtonen [2006] observe that desired reuse must be pinpointed and required from a project if it is to take place, it does not happen spontaneously. Moreover, they observe, *design for reuse* does not either tend to happen spontaneously, but is usually driven by explicit requirements.

The ‘wish list’ to the project, regarding what it should reuse may be formulated in several manners, depending on the desired level of autonomy:

- requirements that particular solutions must be reused;
- constraints on the generated internal variety metrics (e.g. number of new unique components)
- degree of reuse of selected classes of solutions as optimisation criteria
- requirements that the project must actively consider reuse of certain solutions and justify when new solutions have been introduced instead of reusing existing ones. This practice has for example been observed at Toyota, where *“engineers who want to make major changes must provide hard evidence through data that the new design is a major improvement over the existing design.”* [Ward et al 1995 p.51]

Considering reuse in the ‘front end’

Product requirements are often formulated prior to the start of a development project, during a phase often called ‘front-end’. One may ask: when should the front-end explicitly require the project to design by reuse and design for reuse?

The front-end should prepare for the execution of product development projects by interpreting and articulating perceived customer needs, if these are not already specified by the customers themselves, and formulating project and product requirements that are in line with the desired business goals. Problems often seem to arise because of the difficulty in translating business goals into product guidelines, and in incorporating broader business considerations (like impact on product families and supply chains) while defining products and projects. Khurana and Rosenthal state that

> *“the front end requires extensive information gathering and analysis to facilitate the development, testing, and refinements of the new product concept, but this information is not available in one place, role or function. [Front-end-related] cross-functional decisions require an extraordinary degree of coordination among senior management, project managers, functional managers, and core team members.”* [Khurana & Rosenthal 1998, p.67]

So how do successful companies approach the front end? According to Khurana and Rosenthal,

> *"some companies rely on a formal process to lend some order and predictability to the front end. Other companies strive to foster a company-wide culture in which the key participants in front-end activities always remain focused on the following considerations: business vision, technical feasibility, customer focus, schedule, resources, and coordination."* [Khurana & Rosenthal 1998, p.57]

One of the conclusions from their study is that

> *"companies that were more successful in their approach to the front end [...] were able to link business strategy, product strategy, and product-specific decisions [...]"* [Khurana & Rosenthal 1998, p.65].

Arguably, one of the vital issues that link these three aspects (business and product strategy and product specific needs) is the question of what to reuse.

3.6.6 Reuse opportunities identified *during* development

When reuse opportunities are identified during an ongoing product development effort, the following two questions should be asked:

- Does the reuse opportunity benefit the current project, or is it for future benefit of the company? Sometimes, previous designs are reused automatically because it is in the development team's interest to do so. However, in other cases reuse is not *perceived* as being in the short-term interest of the project, especially if there are no incentives to reuse. The problem with the decision to design *for* reuse is that there cannot be any real-time request (pull) for such reusable solutions because of the time lag between design and reuse, so the initiatives must come from 'visionary' forces within the company.
- Does the reuse action imply a net cost of resources to the project, and if so, how is this to be provided/financed? A problem in this respect is that it often is up to the design team to justify that a (more demanding) generic solution is needed to the benefit of future products, i.e. the design team has the burden of proof, which may discourage even obviously beneficial efforts to design for reuse. (Perhaps, in specific cases, design teams should justify why certain solutions have *not* been made reusable.)

Socio-cognitive factors: attitudes and incentive systems

What are the 'spontaneous' reuse behaviours of different development organisations? Studies have shown that the initiative to reuse often is prevented by formal or informal incentives to design from scratch or cognitive obstacles such as engineers being sceptical about other designer's solutions [Busby 1999]. Furthermore, there is often a desire from the part of engineers to do product-specific optimisations that may be at the expense of reuse. Busby argues that the problems of (absent) reuse are more social than technical in nature. Barnes draws a similar conclusion: *"One of the most significant inhibitors of reuse is lack of incentive. Without incentive, reuse becomes a scavenger hunt, where each reuse customer must bear the full cost of finding, understanding and modifying work products."* [Barnes & Bollinger 1991]

On a more general level, it often seems to be a risk of excessive focus being paid on one-product-at-a-time, at the expense of hindering future projects from benefiting from the project results. That is, companies may act opportunistically regarding design reuse decisions by ignoring predictable long-term consequences. Especially when resources become scarce, short-term interests often get prioritised. There are several possible reasons for this, like incentive systems/pressure on project teams to deliver short-term measurable results, fire-fighting, and human cognitive features such as ambiguity aversion [Repenning 2001]. Fisher points out that *"there may be an organizational tendency to 'start from scratch' in designing a new product,* *perhaps because of the costs of finding and testing an existing component."* [Fisher et al 1999]. It seems plausible that designers tend to reuse solutions from a repertoire of familiar solutions, meaning solutions outside this repertoire only are reused if there are explicit requirements for it.

At the individual designer's decision level, opportunism usually plays an important role as a cognitive aspect. For example, opportunism may tell a designer to change the order of the 'reuse tasks' between reuse situations, which has implications for how design support tools should be designed. (Such 'reuse tasks' may be: problem analysis; retrieval, understanding and selection of reusable objects; adaptation; evaluation, etc.) [Sen 1997]

3.6.7 Conclusions

How can the choice of project-heavy vs. functional-heavy organisation affect reuse?
In many cases, the project paradigm does not facilitate reuse, because it emphasises on the product as sole deliverable, and not 'bi-products' such as reusable solutions, and 'bi-effects' such as increased internal variety costs. Project requirements are often much focused around functional requirements, at the expense of other considerations such as reuse needs.

How can the decision structure affect reuse practices?
For those reuse needs that are identified before a development project starts, typically the decision making takes the form of formulation of requirements, constraints, guidelines and optimisation criteria to the development project by the front-end. Reuse opportunities identified during the development effort, are either handled explicitly by senior engineers, 'product boards' or alike, or implicitly by incentives and attitudes in the development teams. In some cases, it may be appropriate that product boards with cross-functional and strategic representatives decide what is allowed to be variant-specific and what needs to be reused/reusable.

4 Trends in product development strategies from a reuse perspective

In the previous chapter, we have analysed design reuse from a number of viewpoints and detail levels. We have found mechanisms (drivers, enablers, inhibitors, etc) and highlighted ways in which these may interact. We have found that one of the most difficult conceptual issues regarding reuse is the decision making under uncertainty about future products. In this chapter, we discuss how these mechanisms relate to current trends in product development strategies. Especially, we focus on two product development strategies or 'philosophies' that have received much attention in the latest years: *product platform* strategies and *lean design*. The intention is to learn how these two strategies affect the reuse practices, not to evaluate or compare the strategies per se.

The questions we aim to answer in this chapter are:

- What are the elements of a product development strategy and how do they relate to the reuse practices in a company?
- Which 'conceptual tools' are there to handle uncertainty in product development?
- What are the 'platform' and 'lean' product development approaches? How do they approach design reuse and what knowledge gap is left?

4.1 Product development strategies

In this section, we explain what we mean by 'product development strategy' and discuss how the reuse approach and uncertainty handling of companies may be affected by it.

4.1.1 Elements of a product development strategy

A product development strategy is basically the way a company intends to manage development of products over longer time (many product generations). Product development strategies should be tailored to specific companies to match their market environment, company culture, material and immaterial assets, etc. Some of the issues that differentiate development strategies are:

- Structure of development organisation
    - project-heavy vs. functional
- Decision structure: top-down or bottom up?
    - In a pure top-down organisation, top management decides on the high-level objectives for the development of products, and then passes orders down the company hierarchy to break down and implement the strategy.
    - In a pure bottom-up organisation, top management creates the right atmosphere for 'grass-root' initiatives to contribute to a positive long-term evolution of the products. Such strategies can be seen for example at smaller high-technological firms that are driven by the engineers' expertise.
- Suppliers and outsourcing policy
- Portfolio management
    - How are the products to develop selected?
    - How extensive are the product roadmaps regarding detail and time span?

- Amount and type of investments in future technologies and processes
    - Fire-fighting or preparing for future projects?
- Handling of product requirements
    - Is development driven by specific customer orders, or by own plans/roadmaps?
    - How are production and other supply chain considerations taken into account during design (as explicit requirements or/and by involvement of cross-functional experts)?
- Handling of uncertainty
- etc

4.1.2 Portfolio management

Portfolio management is one the most important decision areas of a product development strategy, and accordingly, it is often the responsibility of top management. It is also the decision area of most impact on the long-term reuse patterns of a company. The purpose of portfolio management is to choose which products to develop so that company capabilities are linked to customer needs/market potential in the most profitable way. This includes choosing the frequency and innovativeness of product launches using the industry clockspeed as reference.

Krishnan and Ulrich [2001] have identified the following decisions as determining the product strategy:

- What portfolio of product opportunities will be pursued?
- What is the timing of product development projects?
- What, if any, assets (e.g. platforms) will be shared across the products?
- Which technologies will be employed in the product(s)?

In addition, it should be decided which products to *remove*, which is often needed to reduce costs and free resources. These decisions should be based on as much solid data about the present as possible, complemented with forecasts of the technological and market trends and a conscious handling of uncertainty and risk. The results from portfolio management are normally visualised in ‘roadmaps’ which show the introduction and market lifecycle of products and product features.

If a product portfolio changes too fast, companies risk growth problems, such as creating an excessively complex and costly product portfolio. If a product portfolio evolves too slowly, competitors may be given time to develop better and/or cheaper products. Ideally, the product portfolio should be ‘lean’ and change at a reasonable speed when and where it makes sense for business.

Product roadmaps may be more or less detailed and committing. If a roadmap is both detailed and committing, it means that the company can optimise for the forecasted scenario (e.g. the group of expected variants) in order to make the most profit. Of course, if the forecast turns out to be erroneous, the company will suffer the consequences in the form of lost unexploited investments and inflexibility. Therefore, if the future situation of the company is uncertain, roadmaps should be more ‘loose’, signalling it might be wise to invest in flexible assets. The choice can be seen in analogy with how much ‘momentum’ the company is willing to gain. If the company is certain of the goal, the more momentum, the better. But if the company wants to maintain flexibility to change direction as opportunities arise, then the less momentum the better.

These issues, among others, are discussed thoroughly by Cooper [2001] in relation to the well-known Stage-Gate system for managing new product development projects. Roughly, the Stage-Gate system is about dividing the new product development process into a number of consecutive stages, each of which is ‘screened’ when completed and a go/no-go decision is made whether to move forward to the next stage or ultimately kill the project. Cooper proposes five stages which he labels

1. Scoping
2. Building business case
3. Development
4. Testing and validation
5. Launch

Cooper [2001] further proposes a “six F’s” checklist to support a sound evolution of the product portfolio:

- Flexibility: New development projects should be treated differently according to their inherent risk level, with more formalised processes and stringent controls the higher the risk involved. New product development processes are essentially risk-management-processes.
- Fuzzy (conditional) gates: In some cases, stage gate decisions may be allowed to be conditional. This means that the project can be allowed to move forward to the next stage while waiting for a missing piece of information. This information, when it becomes available, may result in the project being reviewed again and possibly stopped.
- Fluidity: Activities may be allowed to span over stage-gates, and stages may overlap. For example concept e.g. be commenced in advance or refined after the main stage where it belongs, if it contributes to the fluidity of the project.
- Focus: Companies should do an effective *portfolio management* focusing resources to the most needing and promising projects. This means that when projects are screened, they should not be considered individually, but in relation to the status of other ongoing projects.
- Facilitation: Somebody should have the role of process facilitator or referee, ensuring gate meetings and other elements of new product processes are followed satisfactorily.
- Forever-green: New product processes should be constantly renewed, to incorporate experiences and adapt to the companies’ unique and changing needs.

In the next section, we investigate what an approach to reuse is, and which the main choices are.

4.1.3 Reuse approach

By *reuse approach*, we mean the pattern of decisions that leads to the design reuse that ‘navigates’ the company in the technological landscape. Such an approach can be more or less conscious and more or less proactive. The reuse approaches of comparable companies can be discerned by looking at factors such as

- the amount reuse that actually takes place, as shown by the companies’ product history
- how much the company invests in preparing for future product development, especially the amount of generic and solution-specific investments in reusability
- when and how design reuse instances are really decided, from the moment of original design to the moment of actual reuse

The selection of design solutions that get reused is determined by how the company answers three questions, explicitly through a product roadmap or implicitly through other decisions:

1. What to reuse from the past?
2. What to share across concurrently developed products?
3. What to make reusable for the future?

Roughly, the two first questions deal with optimising performance and direct/indirect costs where the product requirements and production and logistics capabilities are known. The last of the questions, what to make reusable for the future, deals with the proactive aspect, and the answer needs to be based on forecasts of future needs.

Finally, there are two key management issues that shape the reuse approach

- whether the decision to reuse should be made by individual projects (secondary to functional requirements), or if it should be required explicitly from the project
- how to incorporate (decide and provide resources for) preparation for future reuse into the project duties

Being proactive about reuse

As previously discussed, often companies benefit from being *proactive* about design reuse, which means investing for future reuse. Investments for reuse can be

- Generic, to improve the overall efficiency of reuse. For example, decrease the transfer costs through use of IT-tools for product data management and better documentation routines. Such investments can be beneficial regardless of predictability of specific reuse needs.
- Solution-specific: these are investments related to increase the reusability of particular design solutions. Such investments are normally only beneficial if the solutions eventually get reused.

Generic investments in reuse can have a very positive effect on the reuse practices at companies. For example, it can be the case that by investing in generic design information tools and routines, the reuse possibilities and effects become so visible for decision makers that their decision quality is greatly improved.

Determining how proactive a company should be with regard to the reuse process is a central product-strategic decision. The decision can be viewed in analogy to push and pull systems, which can be defined as follows. A push system

> *"completes a predetermined quantity of work from an established work queue or forecast. Typically, the work queue of forecast is offset to the actual customer demand to allow time for production and delivery."* [Fiore 2005 p.17]

By contrast, a pull system

> *"completes a quantity of work that is directly linked to customer demand. Materials are staged at the point of consumption. As materials are consumed, signals are sent to previous steps in the process to pull forward sufficient materials to replenish only those that have been consumed."* [Fiore 2005 p.18]

Clearly these definitions relate to materials flow, but we see that we can relate them to flows of design solutions as well. Note that we need to distinguish between being proactive about the process and being proactive with respect to the flow of individual design solutions. In the design reuse process there can not be a real-time pull for previous solutions at the moment of reuse, because these have already been designed. But we could say that if the reuse is allowed to be optional we have a pull system, while if the reuse is predetermined we have a push system. In other words, a 'push' reuse process would push design solutions into the future, while a 'pull' reuse process would generate reusable design solutions that in the future can be pulled into that day's products.

Being selective about what to reuse

By selecting design solutions to reuse and preparing others for future reuse, a company controls the flow and evolution of design solutions over time. It is important to note that the rates of evolution typically are different for different product features and supply chain assets. Therefore, reuse planners should take the *relative* maturity of the solutions into account when selecting what to reuse and what to innovate. To do so, the potential rate of evolution of different solutions should be projected against the rate of new product launches. For example, some solutions may become obsolete by the time of there is an opportunity of reuse them, or they may need an enhancement. Other solutions may be sufficiently stable to 'freeze' them and encourage 'mass reuse'. The relative maturity and projected evolution of a design solution should also be assessed against competitors' alternatives, to identify where reuse can provide competitive advantage. Meyer argues that

> *"a company must examine its product architecture to identify those particular subsystems which have the potential to unique proprietary technology and production... Such key subsystems can drive the entire portfolio and should therefore be standardized across it. Once identified, management can pour resources into those highly leverageable subsystems and make them powerful assets of the corporation."* [Meyer & Lehnerd 1997 p121]

Some solutions at the technological forefront may need continuous improvement to remain competitive, and hence must not be standardised. Otherwise, if they are reused unmodified, it can lead to rigidness and stagnation. Therefore, *"an effective approach towards standardization is one that is highly selective, carefully choosing elements that should be standardized."* [Meyer & Lehnerd 1997, p.121]. This may be especially so in turbulent surroundings, as pointed out by Miller [2001]:

> "*In an unstable environment the likeliness of success is bigger when the system consists of stable sub-systems. Development efforts should be focused on front-edge modules while re-using the rest."*

Fiore [2005] suggests that the assortment of parts can be sorted into a range between the two following extremes

- commodity parts and consumables, which typically is the largest category by volume, and contains parts of less complexity such as nuts, bolts, screws, diodes, lubricants, etc
- product-specific parts, designed by the company, typically of higher complexity and lower volume

Companies should identify preferred parts within the above range, i.e. those which the company benefits the most from reusing. Fiore argues that generally, companies should strive to increase the share of preferred commodity parts and consumables, to decrease unnecessary variety.

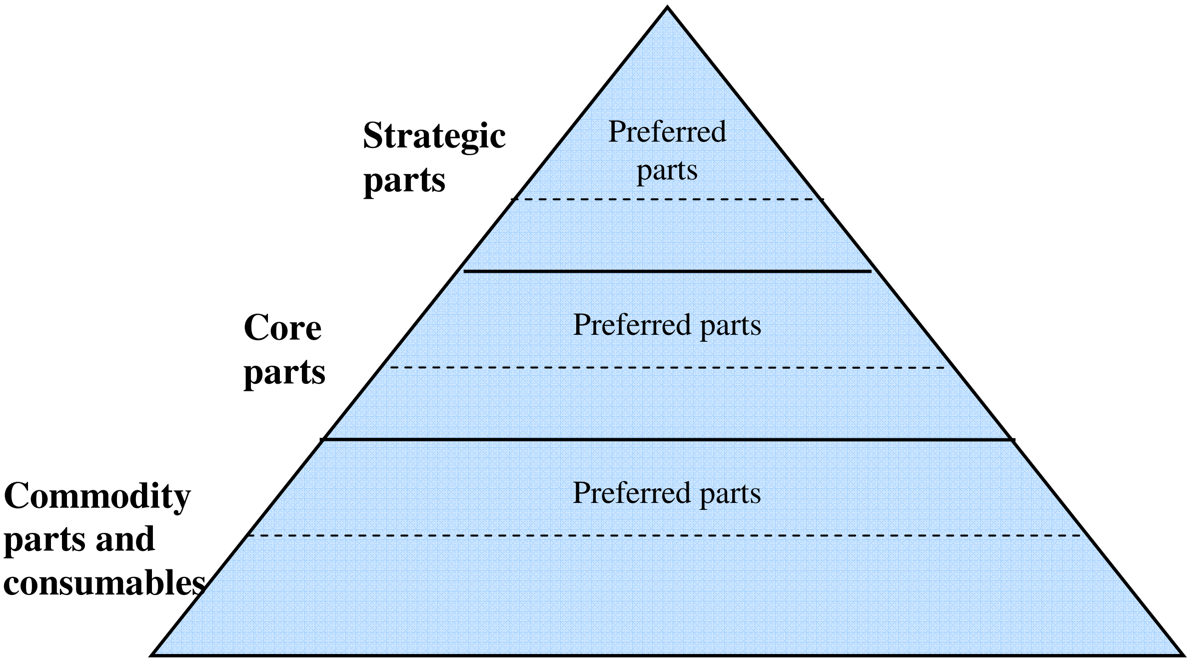


Figure 27: Classification of parts assortment [Fiore 2005]

In the figure, the product-specific parts have been subdivided into strategic parts and core parts, where the strategic parts are those proprietary to the company that give it technological advantage over competitors. Fiore provides an example of how this 'hierarchy of reuse' can be used to guide design reuse choices:

> *"Consider the creation of a new product requiring a shaft component. Using a hierarchical approach, the first choice would be to select a preferred part for use in the new product design. However, if no preferred shafts were acceptable, the second choice would be to select and existing non-preferred shaft. From the company's perspective, this is still better than creating a new part. Lastly, if no existing non-preferred parts were acceptable, then the only recourse would be to design a new part."* [Fiore 2005 p.103]

The same reasoning about physical parts may in general be applicable to design solutions as well. However, if a company is to classify its assortment of design solutions in practice, it will need to restrict the considered solutions to a manageable number by choosing relevant types of abstraction and levels of detail.

Fisher et al propose the categorisation of components according to their influence on customer-perceived product quality. Reuse decisions regarding components with strong influence on customer perception should focus more on performance, whereas decisions about components of weak influence should focus more on cost issues [Fisher et al 1999].

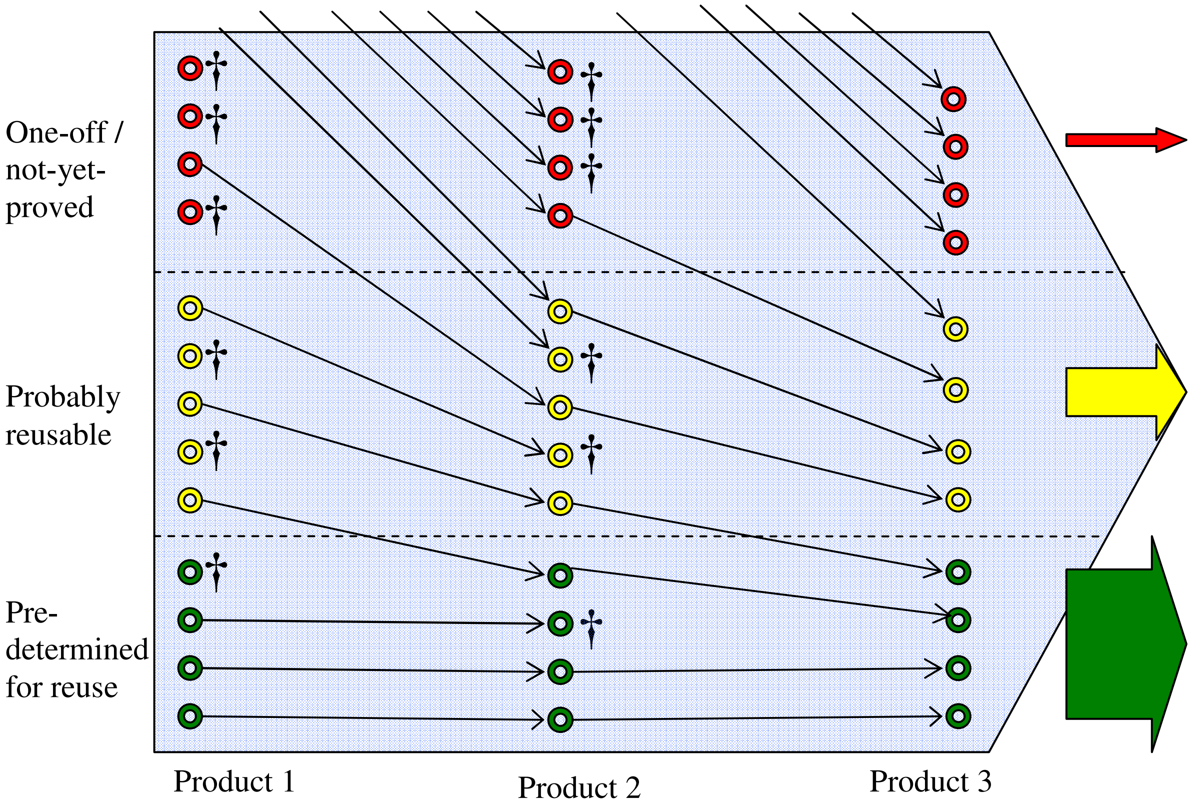
Finally, it is worth noting that the strategic development of company capabilities can be 'directed' by design reuse, both when the reused technology itself is strategically important, and when the reused technology is not. It depends on how the reuse affects technological learning:

- If a less strategically important solution is reused 'as is' to save development effort, this saved effort can be diverted to research and design of other new strategically important solutions.

- Efforts to *design for reuse* can yield a deeper technological knowledge of the solutions to be reused, thus enhancing strategic capabilities.

An illustration of the flow of design solutions

In the figure below we see an illustration of the flow of design solutions over consecutive products. The vertical axis represents the level of 'trust' in the reusability of the particular design solution, which normally coincides with its level of formalisation. Introductions of new design solutions are illustrated by arrows entering from above. Design solutions are moved to a higher level of reusability (lower in the figure) for example through development for reuse, testing, documentation, and by being proved by use in previous products.



Legend:

◎ Design solution

→ Introduction/evolution/reuse of a design solution.

† Discontinued/obsolete design solution

Figure 28: Flow of design solutions over consecutive products.

This illustration type can be used to visualise different forms of reuse patterns at different companies, and help understand how different reuse strategies affect the evolution of the assortment of design solutions. For example, a company reusing extremely few solutions between products could be illustrated as:

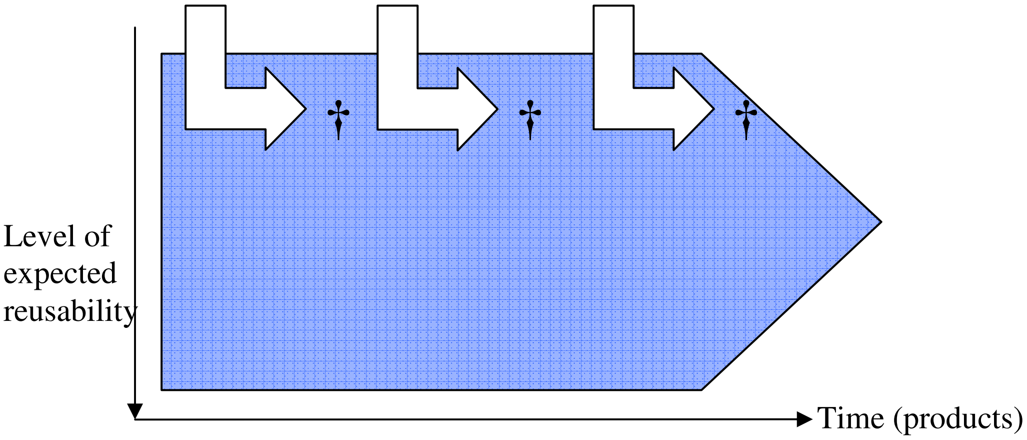


Figure 29: Flow of design solutions with no reuse
The figure illustrates three consecutive products; each of the three bent arrows represent the sum of design solutions that make up each product. In this extreme example, all solutions from a product are discontinued (symbolised by the cross) when moving to the next one. Also, we can see that the solutions never get to the point where there is confidence in their reusability (illustrated by the fact that the arrows remain in the upper area). This can be because reuse is not considered at all, and no efforts are made to facilitate reuse.

4.1.4 Uncertainty handling

The choice of what and how to prepare for future reuse is naturally subject to uncertainties, which can be a significant obstacle because it is often uneasy to act proactively based on scarce and ambiguous data. Therefore, a key to successful reuse practices is to be able to make good decisions under uncertainty. Uncertainty handling in product development projects is complex because processes and managerial decisions interact in *"delayed and nonlinear ways"*, and the effects of the sources of uncertainty are *"not typically cumulative, so strategies designed to manage uncertainty based on one or two conditions alone can be seriously flawed"* [Ford & Sobek 2005 p.175].

Sources of uncertainty
The sources of uncertainty can be endogenous (consequences of design decisions) and exogenous (e.g. budget constraints) [Gonzalez-Zugasti 2001]. Some of the main sources of uncertainty in relation to design reuse are related to:

- the prediction of future products' requirements (confidence, sensitivity to inaccuracies)
- the execution of the preparation (required efforts, likelihood of meeting requirements)
- the effects on the ongoing project (over-design, diversion of project resources)
- the effects on the future projects (likelihood of components reused, savings due to reuse)

Other more general sources of uncertainty can be:

- own understanding/bias
- changes of customer preferences
- actions by other actors (competitors, government, etc)
- cost changes
- new technological alternatives
- changes of own capabilities

Handling uncertainty

Uncertainty can in many cases be considerably reduced through the analysis of existing information like historical data. Such efforts aim at reducing 'unnecessary' uncertainty. The remaining uncertainty can be handled in two conceptually different ways, depending on how much uncertainty the organisation can 'live with'.

- Qualified guessing: One extreme is that teams at an early stage produce estimative values for the uncertain parameters, which then are passed on to the rest of the organisation that from then on treats these values as 'certain'. For example, if there is uncertainty about what size the customers will prefer, say from 1 to 3 of some unit, then a team 'decides' that the customers will want size 2, which is then treated as 'fact'. The benefits of this approach is that it allows more 'momentum' and requires less coordination, and the drawback is of course the risk that the estimation turns out to be inaccurate, leading to costly late design changes or lower product quality.
- Postponement: The other extreme is that the organisation works on with uncertainties and tries to be flexible to adapt as well as possible to any outcome in the expected uncertainty range [Yang et al 2004]. This is for instance what Toyota's set-based approach is [Ward et al 1995, Sobek et al 1999]: Toyota's engineers proceed with designing based on a specification *range* or set, until more information allows them to narrow the specification set with greater accuracy. For example: "begin designing part x that shall have a length y which will fall into the range 2-3cm". The benefit of this approach is improved design decision accuracy, and the drawback is potentially lost development momentum, as the ambiguity may cause increased need for coordination efforts.

These two strategies can be described as early-converging vs. late-converging, respectively. In early converging strategies, an alternative is chosen immediately, while in a late converging strategy, the choice is made later [Ford 2005].

The 'attitude' upon ambiguity or risk aversion policy, can be of different types:

- aggressive
  - investing heavily despite risks in pursue of high future benefits
- defensive or risk-aversion policy
  - preferring not to invest for ambiguous outcomes
  - minimizing the risk of failure at the expense of benefits of success
  - can be about investing to minimise risks
- explorative/agile
  - lightly investing in pilot projects, qualified trial and error style
  - can be very effective if a company is good at learning
  - example: Helly Hansen tries many product concepts 'lightly', and then senses the market and invests in mass production when a product appears to be successful

Technology selection under uncertainty

Concretely, uncertainty handling with regard to design reuse is often about choosing the right technologies. Krishnan and Bhattacharya have studied this situation, by modelling the choice of reusing a previous proven technology versus introducing a new less proven technology with a potentially higher cost/performance ratio [Krishnan & Bhattacharya 2002]. They have identified three main alternatives:

1. *"pizza bin approach"*: rejecting the new technology without consideration;

2. *“parallel path approach”*: starting the parallel designing of two variants of the product, one with the new technology and one with the old, until the insight gained during the development shows which one should be aborted and which one should be completed;
3. *“sufficient design approach”*: (over-)designing the product so that it can function both with the new technology and with the old;

In alternatives 2 and 3, a *flexible design approach* is pursued, where the decision of which technology to commit to is postponed to a later stage in the project. According to their model, the best alternative should be chosen with consideration to

- the assessed viability and uncertainty level (variance) of the new technology before the start of the project
- the expected impact on profits if the new technology is successfully integrated in the product (increased sales and/or less production costs)
- the impact on profits of the *development cost* for each technological alternative
- the impact on profits of delaying time-to-market
- the company’s risk aversion

If the parallel path approach is used, the selection of technology should be made *as soon as possible* as new knowledge yields a *negative* calculated net benefit for any or both of the technologies. Pursuing parallel paths has a high development cost that increases as the project advances.

If the sufficient design approach is followed, a larger development effort is invested and ‘locked’ from the beginning, compared to designing for a single technology. According to the model, in the sufficient design approach there is no cost for postponing the selection of technology, so it can be advantageous to select as *late as possible,* or rather, as late as needed to be very confident about the viability of the new technology [Krishnan & Bhattacharya 2002].

4.2 Platform-based development

In a textbook product platform strategy, a product family architecture (PFA) specifies a set of generic design solutions (often subsystems) that are reused in a series of product variants. The PFA is optimised for variety and commonality taking into consideration the whole supply chain. A central point of product platforms is that the generic solutions are deliberately decided in advance upon the designing of the PFA. Once the PFA has been defined, less designing and coordination effort is required to create product variants, since much of the work will be laid out by the design rules derived from the PFA. Note that within the design areas that are ‘outside’ the PFA and thus delegated to variant-specific design (‘differentiating attributes’) reuse is optional.

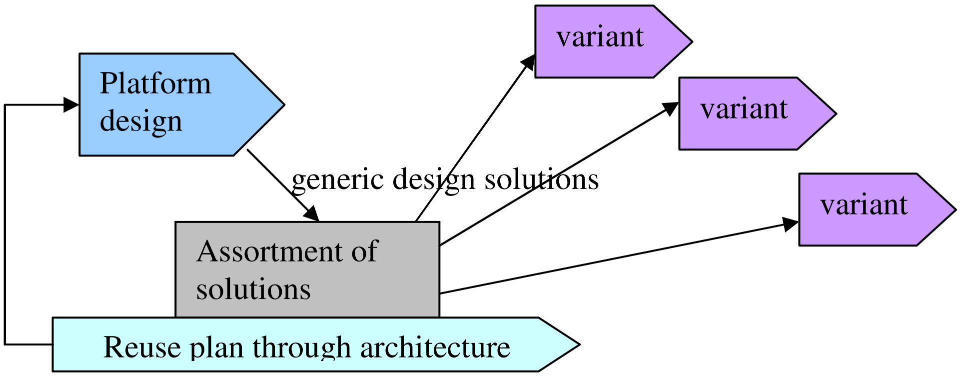


Figure 30: A platform approach; the decision to reuse is made a priori

Below follows a review of current literature on product platforms that aims to show theoretical and practical views on product platforms.

4.2.1 Goals and scope

Product platform strategies aim to exploit commonality between members of a product family while providing an attractive variety to the market, i.e. a co-optimisation of variety and commonality [Berglund & Claesson 2005]. Ultimately, the purpose of platform strategies is to achieve an efficient use and development of the company's core assets through an alignment between the products' designs and the supply chain, to gain a *competitive advantage* [Kristjansson & Hildre 2004]

The main benefits are expected to be:

- reduced marginal cost and development lead time for variants
- reduced costs for the product family as a whole

One often cited example of a successful product platform strategy implementation is the one of the Sony Walkman, where Sony got to be extremely efficient at developing new product variants extremely frequently and cheaply, thus keeping Sony ahead of competition for many years [Sanderson & Uzumeri 1995].

In addition, some companies seem to adopt platform strategies in pursue of company rationalisation and alignment, in the same manner as many companies once embraced Total Quality Management - TQM. Specifically, platform strategies are then hoped to help them exploit their *core capabilities* maximally.

Enabling factors

In order for product platform strategies to be successful, the core technology of the platform has to be stable and aligned with market needs to remain competitive during the period when the derivatives are to be created, until the platform itself is upgraded. Especially, the functional architecture must be stable [Miller 2001]. Product platform strategies are thus especially applicable for companies operating in markets where customer requirements are fairly predictable and the core technology is relatively mature. Note that platforms may be ideal to use certain stable technologies as a base for coping with other unstable technologies.

The match between company capabilities and market situation and the challenges ahead are also highlighted by Juuti et al:

> *"Succeeding in platform engineering calls for critical mass of people in communities of practice... Effective utilisation and description of knowledge as a valuable, manageable*

> *asset is yet to be discovered. This implies further integration of financial aspects and processes used for assessing the benefits of platforms."* [Juuti et al 2004].

When a product kind is very mature and produced in high volumes, cost aspects become so important that it becomes necessary to optimise each variant individually. This makes it difficult to find a satisfactory platform solution to share between variants. This is due to the high volumes and hard cost-competition that make the unit cost more important than for example development lead time.

4.2.2 Means

To achieve these goals, the key recipe of platform strategies is to explicitly differentiate what is to remain invariant across the planned products and what is to vary to provide the desired variety to the market. This can be about parts of the products and/or parts of the supply chains. A development process is decided for the entire product family development process, which is to apply the designing mindset based on co-optimisation of commonality and variety.

The first implementation step is to make an analysis of the company's market situation in order to choose which product *variety* the company should provide to its customers. Meyer suggests that companies should identify market segments and 'price-performance' (low- to high-end) tiers, perhaps most relevant in relation to consumer products [Meyer & Lehnerd 1997].

Second, goals for *commonality* should be chosen. This is normally done by identifying areas where variety has a high cost to the company but a low value to the customer.

Third and last, the product and processes architectures should be designed to exploit the benefits of commonality (usually economies of scale and low complexity), while paving the way for an efficient creation of variety.

We can distinguish between product and process platforms. In a *process* platform, the production and other supply chain processes are standardised, that is, these processes determine the commonality and variety optimisation criteria. Process platform strategies provide variety to the market by varying the design parameters that do not affect production efficiency. For example, casting processes can allow a wide range of product designs, and inexpensive alternation between these.

Mortensen et al [2005] argue that many companies of today are making a transition from one-at-a-time product development processes to 'multi product development', in which product platform-based development they claim is a vital tool. They propose a 'platform maturity' scale of six levels, based on nine characteristics which companies must change in order to succeed with the transition from single– to multi product development. The intention is to highlight the transition concerns not only product development but all functional areas of companies. The characteristics are:

- *Procedure*: Product development should not be executed 'in one go', but separated into preparation and execution, much in the same way that manufacturing is prepared before executed.
- *Building principle*: Companies move from product-by-product constitutive models (i.e. the description of physical parts and their interconnections, or "structure") towards product assortment-based constitutive and behavioural models (or "architecture")

- *Product plan*: should move from single-level to multi-level, including products in the foreseeable future and road-mapping the introduction and expected useful life of product features and technologies over several products.
- *Organisation*: Should move to three-level development, covering architectures, standard designs and product projects, respectively.
- *Business evaluation*: Not just variable costs should be considered when choosing product solutions, but the total costs/benefits for the company.
- *Reuse measurement*: Should be introduced to stimulate effective asset reuse.
- *Building principle responsible*: Management teams should take responsibility for choice of building principles.
- *Business processes related to building principle*: Companies should align all business processes to *harvest* the benefits of the chosen building principles.
- *Formalisation of building principle*: The building principle should be documented to facilitate its communication.

What is a 'platform'?

While there appears to be consensus about what the *mindset* behind a platform approach is, there are differing opinions about what one *platform* itself is. Some view platforms more loosely as a measure of synergy effects between developed products, thus implying all companies would benefit from platforms. Others have a narrower view that sees platforms as a tool among many that may or may not suit particular companies. Some opinions of what a platform is are:

- a set of design solutions (subsystems and interfaces in assembled products) whose reuse is obligatory in each product variant
- core capabilities/resources that are exploited maximally

Meyer and Dalal [2002] define a product platform as *"a common subsystem or subsystem interfaces that is leveraged across a series of individual products by means of shared product architecture"*. In this view, the product platform is the most stable and reused of the subsystems across a product family, and the method aims at designing it as well as possible and making the best use of it. This narrower view, where the concept of platform is associated with the product architecture and technologies, appears to be quite predominant in industry, as suggested by a multiple-case study by Halman et al [2003].

In another view, a platform is *"the collection of assets that are shared by a set of products"* (at a given time point), where the assets include components, processes, knowledge, and people and relationships [Robertson 1998]. Such an use of the term appears to be more spread in academia than in industry. This view focuses more on the product development process, and regards the platform strategy as a reuse/renewal policy related with the company's core assets. The view is rather broad, and is practically applicable to any company. In other words, this view does not regard the platform approach as a tool among other tools, applicable to some companies but not other, but rather it sees it as a measure of company excellence – the more the better. For this view to be operational, the company must elaborate action plans to enhance its platform leverage.

Product family architecture design

A product family architecture (PFA) can be of various kinds, typically combining modular, standard-based, and scalable/parametric solutions. The PFA determines what is shared across variants and what differentiates the variants. It should be optimised for the performance/cost ratio of the product family as a whole, including the time-to-market-cost for new variants [Meyer

& Lehnerd 1997]. Many contributions have been made to the issue of how to design the PFA. For example, Johannesson and Claesson [2005] propose a procedure for PFA design supported by tools that model design solutions both at a functional level and at an embodiment level. In their approach, the product structure is built up by parameterised ‘configurable components’, which allows for an efficient subsequent configuration of products (i.e. creation of variants). Muffatto and Roveda [2002] have studied how the PFA affects the opportunities and constraints for platforms and Simpson [2001] has proposed a method for designing a scalable PFA.

The design of the PFA can be very demanding. Miller argues that modular platform systems are more difficult to design than comparable non-platform systems, because the interactions between modules must be specified in advance, and that mature architectural knowledge is needed, covering

> “*the complex web of knowledge about functioning, inner workings, and interdependencies of the technical systems, knowledge about the realisation processes, sales, distribution, and use, and knowledge about the business, technological trends, needs and strategies.”* [Miller 2001]

One could argue that the same knowledge is needed also to develop products one-by-one. Perhaps the difference is that often a new PFA will raise the stakes, making it more costly to fail, and will be more difficult to correct between product variants. Therefore, many design flaws that the company could ‘live with’ in the short term if developing products one-by-one, are simply not allowable in a PFA. This pressure to achieve high design quality in the PFA may give the company increased insights about its operations, and make problems visible that otherwise could pass inadvertent.

4.2.3 Platform-based organisation of product development

While ‘platform-thinking’ is a mindset that may be applicable to many ‘micro-scale’ situations, most platform strategies as discussed in literature (e.g. [Meyer & Lehnerd 1997; Gonzalez-Zugasti 2001; Krishnan and Gupta 2001]) are top-down in nature and require company-wide realignments. For example, Meyer and Lehnerd argue that the success of the power-tool product platform to a great extent was due to the *holistic* approach:

> *“Black & Decker did three things right:*
>
> - *It avoided piecemeal, single-product focus. Instead, management dealt with the power tool product line as a whole.*
> - *It bridged the traditional divide between engineering and manufacturing with the result that both products and the processes for creating them were simultaneously redesign.*
> - *Senior management adopted a long-term horizon and made the initiative a top priority.”* [Meyer & Lehnerd 1997, p15-19]

At the heart of platform-strategies is the ‘prepared derivative creation’ or ‘procedure preparation’ that makes the execution of product variant development projects more effective [Mortensen 2005]. In line with this, it is often convenient to separate the development of the shared subsystems and interfaces from the development of variants. By separating the activities of derivatives creation and platform development, often more resources can be allocated to the latter. However, separating the activities may have other effects, positive and negative, that may not be obvious. For example, it may impact on the interests of project managers and line managers, and may make it easier to allocate responsibilities for long-term development to platform teams. Interestingly, Muffatto [1998], finds that

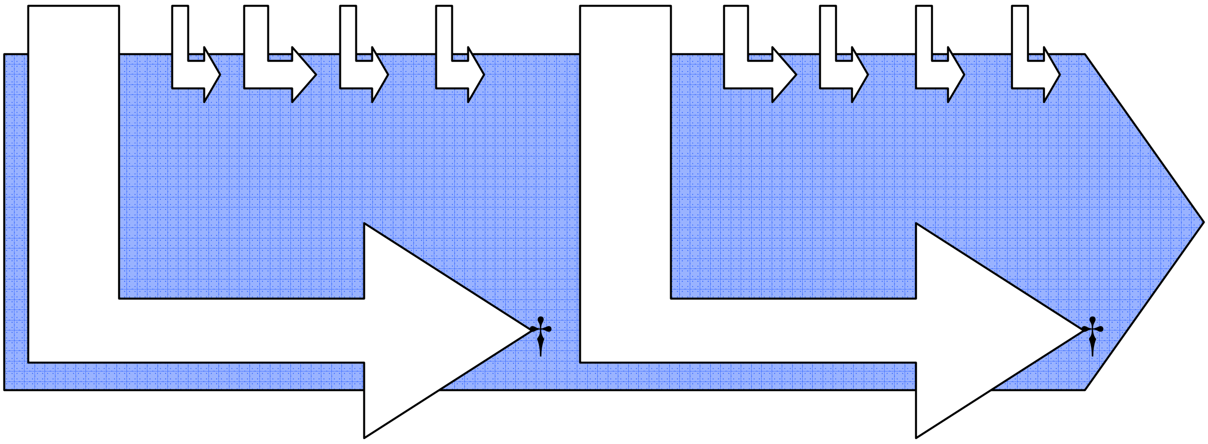
> *"the introduction of a platform concept is also influenced by the company's development culture. For example the traditional high degree of autonomy among Honda work groups makes sharing such an important part of the vehicle more difficult. The same is true for Toyota, where the presence of powerful project managers makes difficult a full adoption of the platform concept."*

Platform and variant development have different immediate customers. Those working with platform development determine some design rules for those who are to work with derivative creation (the later will not be allowed to change the product platform inside a given time/project scope).

Platform development activities should be more focused on research of the underlying technology, as a long term trend, because the platform lifetime should survive the lifetime of several variants. Platform design is often technologically challenging while having very high quality requirements, because design problems may impact the whole product family. Therefore, platform development projects normally have long lead times. Requirements include non-functional and holistic quality properties such as robustness, designer friendliness, portability, future compatibility, etc.

Variant designing should be more agile and sensitive to marketing needs, has focus on minimising market risk and does not need to focus so much on the technological risk, because a product variant can be replaced in the market easier than an entire product platform. The requirements are typically focused on the requirements of targeted customer niches, following the platform design rules that ideally should ensure manufacturability and other cost issues.

A company making derivatives of a static product platform could be illustrated as follows, if we exaggerate the flow patterns to the extreme. In the figure, we can see that there are two kinds of solution flow: the design solutions constituting the platform itself (large arrow), which is from the beginning prepared to be reused (lower area of the figure). The variant-specific solutions, which may be mere *configurations* of the platform solutions (small arrows at the top), are designed 'from scratch' for each product and never reused. Of course, in reality this situation will never be as extreme as the figure suggests. Reuse will often take place between consecutive platforms (which is one way of ensuring its stability, even if they represent mayor architectural innovations) and even variant-specific solutions will be at least partially reused.



The large arrow represents the platform which in each product variant is reused in combination with variant-specific solutions (small arrows)

Figure 31: Platform type of reuse flow, over two platform generations that are designed from scratch.

4.2.4 Challenges

One issue that appears to be very challenging for companies that have adopted platform strategies is how to *renew* product platforms. This issue is discussed below, followed by some remarks about some remaining knowledge gaps in literature.

Renewing platforms
Renewing platforms often poses a considerable challenge. One reason for this is the 'inertia' that a platform gathers because the development organisation has a tendency to get 'accustomed' to the promise that the platform will not change. Therefore, many product and production solutions are allowed to *depend* on the particular platform implementation of the time, which make them difficult to reuse if/when the platform is to be redesigned.

Meyer claims that companies often start from scratch with a new product platform rather than continuously renew an existing one. This can be a very costly and risky business, and a sign of how difficult it can to make changes to a widely reused set of design solutions. If a shared component is modified, it may be necessary to execute costly testing in all the products using it, or risk quality problems, if the company has not an efficient process in place supporting this. Therefore, some companies may prefer not to renew a platform incrementally and instead save all improvements to the next platform generation, which may have negative impact on competitiveness. Meyer is one of many advocates of continuous platform renewal:

> *"To achieve sustained success in new product development, a firm must continuously renew its platform architectures and their manufacturing processes by integrating advances in core product and process technologies... Renewal is balanced between core product and process technology development, the integration of these technologies into successive generations of product platforms, and the creation of specific derivative products for both existing and emerging market niches. By consistently obsoleting its own products with better ones, the company keeps the heat on its competitors and ensures the perpetuity of the enterprise."* [Meyer & Lehnerd 1997 p.37]

Kim [2003] has studied the transition of 'technology lifecycle', which in many companies corresponds to renewal of product platform solutions. In his study, Kim aims to answer

> *"how the optimal switching time between technology generations is determined by such factors as technological uncertainty embedded in the technology, the cost to switch from one generation to another, and the utility which the economy enjoys by utilizing the technology platform."* [Kim 2003, p.371]

Naturally, there are no exact answers to such questions, but we note that there is a variety of tradeoffs involved, in which the uncertainty dimension has a central role.

There is not as much support in literature for how to achieve platform renewal, as there is for planning and designing a product family from scratch. One of the contributions that do incorporate renewal, however at a very conceptual level, is Dynamic Modularisation. Lehtonen discusses this issue and the potential of Dynamic Modularisation as follows.

> *"If we claim that platform-based modular structure could cope with customer variation in mass-customising paradigm, there still exists one challenge: the variation within the product family life-cycle. We have earlier presented Dynamic Modularisation (Dymo) business and product development paradigm, which adds the platform-paradigm the company processes that are needed in handling the life-cycle variation* [Riitahuhta 2001]. *The core idea of Dymo is making product development work on two levels. On the upper level there is platform development. This includes the customer requirement management, product architecture management and development and module creation process, where suitable modules are developed for fulfilling customer requirements. All these actions are targeted for creating a product platform, which enables launching a product family, which corresponds to market needs now and in predictable future."* [Lehtonen 2003]

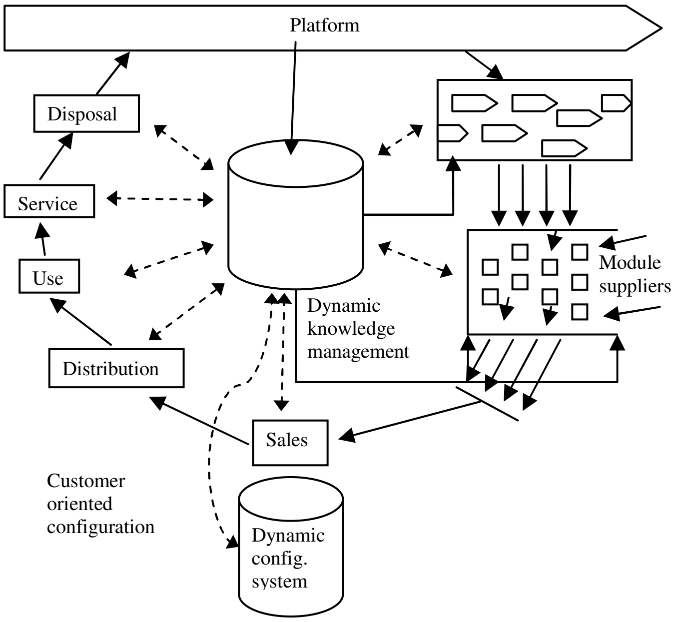


Figure 32: Dynamic Modularisation [Riitahuhta 2001] (see enlarged figure in chapter 3.1)

Dynamic modularisation is a concept that covers the gradual evolution of the product portfolio with no time limit. This is a very interesting contribution to our research, because reuse management is incorporated in a continuous process. It is assumed that the same mindset can be applied also to non-modular systems, as we return to in the conclusions of this thesis. Our results are in line with the dynamic modularisation approach, only we are more detailed and specific regarding *design reuse*. In the figure above, it is the higher right corner that we are focusing on, that is, the process of selecting modules (design solutions) to reuse and mix them with new ones from module suppliers (new development efforts).

Knowledge gap
Halman et al have identified ‘white spots’, where the literature on product platforms do not support the needs in companies that have adopted platform strategies. For example, they claim the need for future research to “*develop categories of options for platform and product family development that are useful in practice given a specific context*” [Halman et al 2003 p.13], as opposed to very straightforward success story examples that neglect the difficult issues. Furthermore, there is a lack in literature of guidelines and decision rules to support companies when reacting to technological or market changes, and ‘strategic’ selection between platform alternatives. Specifically, an “*important gap in platform literature ... is the lack of a sound valuation model, as traditional methods (e.g., net present value) fail to provide the necessary support for valuation and decision making*” [Halman et al 2003 p.13].

Simpson presents a general survey of the status quo of product platform research and practice. He concludes that the following areas need further research [Simpson 2003, p.7-8]:
- Product family planning and platform development
- Quantifying the benefits and drawbacks of platform based product development
- Design for Manufacturing and Assembly (DFMA)
- Web-based platform customization strategies
- Support for small and mid-size manufacturers
- Overcoming organizational barriers to platform-based product development

Curiously, uncertainty handling is not directly mentioned, although it is inherent in most of the mentioned issues. The reasoning could be that if sufficient knowledge and tools can be developed supporting the above list of issues, then their inherent uncertainty can be reduced to a negligible level. If that could be achieved, it might be possible and sufficient to base decisions on measurable data alone, and there would be no need to make decisions under considerable uncertainty.

Platform strategies aim at isolating uncertainties in product requirements and design solution properties by designing them into the *variant-specific* parts of the products. The shared platform subsystems themselves will normally be very costly to modify after adoption, so it is important that the requirement uncertainties are reduced to a minimum when designing them.

4.3 Lean product development

Lean approaches to product development, in contrast to the above discussed platform strategies, do not focus on heavy planning and control of an explicitly defined basis for commonality and variety. Instead, lean approaches aim at making the product development process as fluid and self-steering as possible. Lean design relies on skilled, responsible and empowered engineers to make strategically sound decisions and improve the product development process. The goal is to maximise value-adding activities in the company, which is equivalent to minimising waste.

Often Japanese companies are cited as employing the lean approach. Often, this refers to ‘lean production’ which was the label that was given to the Japanese way of increasing their production efficiency to becoming best of the world, and which many American and European manufacturing companies nowadays have adopted. Toyota is the most used example of a successful ‘lean’ company. Toyota has created a company culture that has made them world leaders in product development efficiency in the automobile industry. Toyota’s and other companies’ way of doing product development has similarities with the flow-based mindset behind lean production. Therefore, some scholars have chosen to call this mindset ‘lean design’, or ‘lean product development’, but other labels can also be found, such as ‘The Toyota Product Development System’ [Kennedy 2004].

In many traditional product development organisations, the design process is rather linear. The design team generates specifications based on one chosen system concept, and the product is partitioned into subsystems which then are developed and put together for testing. If problems are encountered at this stage, the team must go back and redo a considerable part of the process. This process can be rigid as focus is put on procedures and compliance and progress is controlled by counting completed tasks. At Toyota, by contrast,

> *“The goal is not to complete a certain number of tasks or maintain a specific production rate, but to generate a constant flow of new products. So instead of focusing on developing a particular device, the company tries to create a steady value stream of new products.”* [Kennedy 2004 p.152]

This is done by generating many alternatives for each solution from a multiplicity of perspectives, and keeping interfaces loose to increase the flexibility to combine solution alternatives as these are better understood. Furthermore, learning is highly prioritised, allowing for reuse of knowledge about which solutions work and which do not.

4.3.1 Lean design principles

The ‘lean’ approach is not as much a strategy as it is a mindset, or a set of principles that can be applied to many different situations. The ‘lean principles’ are [Fiore 2005]:

1. Specify value in the eyes of the customer.
2. Identify the value stream and eliminate waste.
3. Make value flow at the pull of the customer.
4. Involve and empower the employees.
5. Continuously improve in pursuit of perfection.

Lean design has a lot in common with Six Sigma, but should not be confused with it. Six Sigma is a product-based quality approach that uses *“variation reduction, rework and scrap elimination, and process control to improve product quality”*. By contrast, Lean design is a *process*-based approach that uses *“the concepts of value streams, waste elimination, work concentration, and flow to meet the goal of reduced cycle time”* [Fiore 2005, p.8].

In the lean approach, design solutions flow across product generations in a rather steady stream. This design solution flow can be illustrated as follows.

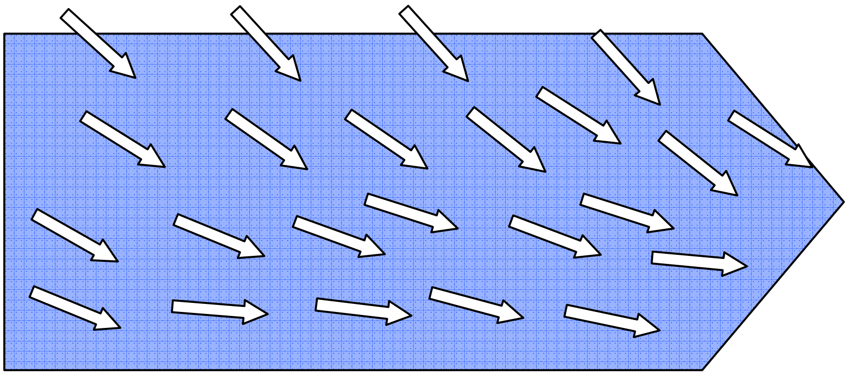


Figure 33: Continuous development-type of reuse flow

As the figure shows, design solutions are continuously improved so they gradually become more reusable. Obsolete solutions are replaced by better alternatives one-by-one, instead of replacing whole clusters of solutions at once.

Lean design approaches appear to handle uncertainty by 'narrowing' design decisions gradually as information becomes available and likely alternatives are explored. Uncertainty handling is therefore inherent in the set-based approach. Uncertainty is incorporated as a property of design issues, rather than first approximating uncertain factors with their most likely values and then using these values as if they were certain in the decision process.

4.3.2 The incremental innovation approach

In a study of Japanese, European and U.S. car manufacturers, Clark and Fujimoto have observed two distinct strategies: the "infrequent great-leap-forward" (fewer but more innovative product introductions), most practiced by American firms, and the "rapid-inch-up" (more frequent but less innovative product introductions) practiced by the 'lean' Japanese firms, with the European firms applying a mix of the two.

> *"In both product and process technology (particularly the latter), Japanese firms attempt smaller technological jumps and tend to limit resource commitment, carrying out chassis and electronics development in existing plants using existing equipment with incremental innovations"* [Clark & Fujimoto 1991 p.135].

Therefore we can conclude that Japanese firms successfully practice considerably much *partial* reuse. The authors further compare the approaches as follows:

> *"The 'rapid-inch-up' strategy of Japanese firms offers some potential advantages over the 'infrequent great-leap-forward' strategy Western firms have tended to follow. Because next-generation designs use established processing concepts, Japanese firms may avoid startup confusion; moreover, regular and frequent changes of technology enable the product development organisation to establish an 'rhythm' of development, streamline the development process, and orient the entire organisation to continual learning and improvement"* [Clark & Fujimoto 1991 p.135]

The avoidance of radical innovations has also been noted in another study that analyses the development process at Toyota. This is because Toyota is aware of the risks involved in disrupting the flow of development, and is confident that sufficient innovation to be attractive to its customers can be achieved incrementally. The authors claim that Toyota is rather selective regarding the exploration of the solution space for different product features, preferring ‘safe’ improvements than ‘creative’ (and risky) ones.

> *“Toyota has a reputation as a conservative company for good reason. Getting a high-quality vehicle on the market very quickly is a top-priority, and Toyota realises that excessive experimentation will make this impossible.”* [Ward et al 1995 p.51]

In Clark and Fujimoto’s comparison between regions, one observation is that U.S. firms tend to reuse a larger share of the components in new cars than their Japanese counterparts. On the surface, this seems paradoxical given that Japanese firms on average have lower fixed development and manufacturing cost per new product, and it is often expected that more reuse should mean lower fixed costs. What happens is that Japanese firms incrementally improve many existing solutions, i.e. they partially reuse, which in the statistics fail to show because an improved component is counted as a new component, even if it still can be manufactured with most of existing tools. In contrast, when the studied U.S. firms chose not to reuse an existing solution, they often radically changed it, because it had not been improved for longer time. This has implications for the amount of development effort devoted to new and reused parts at different stages.

> *“Because it is implicit in the great-leap-forward strategy that a new part is very new (the design of an old part may be very old, 12-15 years in many cases), the decision to use a new part is a decision to engage in significant new engineering work. In contrast, in the Japanese rapid inch-up strategy, the design of each generation of parts is closely related to that of the previous generation, and the resulting emphasis on process continuity holds the promise of lower costs. Producing a ‘new’ part in a Japanese company following a rapid inch-up strategy is thus less of a stretch for both engineers and the capital budget”* [Clark & Fujimoto 1991 p.150]

4.3.3 Set-based design approach

One of the outstanding papers studying lean product development is the one by Ward et al [1995]. In their case study, the authors present findings that at first sight, the development process at Toyota seems inefficient and cumbersome, contradicting the fact that Toyota is one of the leaders in product development efficiency. The main observation is that, contrary to the often ‘preached’ practice of freezing product specifications as early as possible in order to break down the design work into more detailed levels, at Toyota design decisions are taken in sets of alternatives and/or ranges, thus delaying the final decisions until late in the design process. The authors call this approach set-based concurrent engineering (also presented in [Sobek et al 1999]). In the set-based approach, designers spend a considerable effort exploring design alternatives in parallel. The advantage of this approach is that, when the final design decisions are narrowed to a final specification, then the amount of information about the alternatives is considerably larger than was early in the process, thus granting higher quality of decisions and less rework because of design flaws. This is among others explained by the general manager of body engineering, who in the study remarked that

> *“delaying the decision on critical dimensions until the last possible moment is necessary to ensure that customer’s expectations are fully understood, that they will be satisfied by the body design, and that the design is also manufacturable. He described preventing engineers from making premature design decisions as a critical part of his job. In contrast, managers at two U.S. companies told us that the key to achieving manufacturable designs quickly was to freeze the hardpoints early... thereby giving manufacturing more time to respond, but risking a suboptimal solution.”* [Ward et al 1995 p.52]

The set-based approach seems to require higher level of engineering skills than other, 'point-to-point' approaches, as noted by Ward et al:

> "Toyota has unusual engineering expertise. Toyota engineers serve a minimum of fifteen years before reaching management positions, have extensive hands-on experience, undergo frequent training, and are vigorously encouraged to think about their jobs and technologies by managers who are themselves technical experts. Set-based concurrent engineering seems likely to require more skill and judgment than the simpler, point-to-point approach; its adoption by companies with lower average skill levels might be difficult." [Ward et al 1995 p.60]

4.3.4 Organisation of product development

Regarding the organisation of the product development work, the studied lean companies show a less structured approach than comparable traditional western companies. For example, it has been found that Toyota does *not* apply

> *"some of the concurrent engineering mechanisms that U.S. companies consider essential, such as collocated, dedicated multifunctional teams, a highly structured development process and frequent meetings with suppliers."* [Ward et al 1995 p.47]

This seems to contradict the recommendations of much management literature. One possible explanation why this organisation style appears to work for Toyota, may be that

> *"Each U.S. practice incurs costs. Dedicating and collocating teams can, in the long term, degrade the company's technical expertise as specialist organisations disappear or become moribund. Detailed process manuals can be cumbersome. Communicating through frequent meetings is expensive; U.S. engineers often explain they spend too much time on meetings, and not enough on creating or analysing."* [Ward et al 1995 p.47]

It is noteworthy that Japanese automobile companies also have arrived at commonality/variety approaches that resemble product platform strategies, not as a result of large cost-reduction programs, but as a result of design simplification. However, Muffatto notes in a multiple-case study of Japanese automobile companies that, *"platform development is still very close to model development."* [Muffatto 1999, p.152]. Considering their relative success, it appears they have found ways of organising their product development that suit their company cultures well, but need not automatically suit companies in other countries.

4.4 Conclusions

We have discussed two big streams of product development strategies today: platform strategies and lean design strategies. Lean and platform strategies are not mutually exclusive, but rather complement each other. In our view, it is fully possible for a company to adopt a 'lean platform strategy', by choosing a 'lean way' to develop a product portfolio that co-optimises commonality and variety. However, some of the literature on how to *implement* product platform strategies imply a top-down system based on centralised plans and one-way work delegation which appears to conflict with the rather 'smooth' negotiation-rich approach of lean design.

What are the elements of a product development strategy and how do they relate to the reuse practices in a company?

A product development strategy includes a variety of issues of which the ones with most impact on the reuse practices of a company are:

- The type of organisation of product development. Project-heavy or matrix/functional? Functionally-oriented organisations may allow more incentives for transfer of design knowledge between projects.
- Portfolio management. Whether research and development is driven by customer orders or own plans for future will impact the amount of investments in reusability.
- Uncertainty handling. A proactive approach to design reuse should be based on forecasts about the future and provide flexibility to cope with unpredictable developments.

Which 'conceptual tools' are there to handle uncertainty in product development?
The first step in any uncertainty handling must be to reduce unnecessary uncertainty by spending effort to gather available data, according to the amount of time and resources available for this. The remaining uncertainty should then be assessed to conclude whether

- Qualified guesses can be used to provide estimates which can be trusted to the level as to be worked with. This is often a good approach when the uncertainty concerns one or two dimensions, like when estimating needed design work hours for a task, and the consequences of inaccuracies are acceptable.
- Whether the work must proceed with an uncertainty range, until more knowledge is gained (decision postponement). This may be a good approach when the uncertainty regards many dimensions (alternative design principles) and the consequences of inaccuracies are costlier than the cost of postponing the decision.

What are the 'platform' and 'lean' product development approaches? How do they approach design reuse and what knowledge gap is left?

Platform strategies
Product platform strategies are based on the optimisation of commonality and variety for a series of products by means of a product family architecture (PFA) that is designed before the creation of variants. Reuse in platform strategies is predetermined. For example, in modular products, the 'platform' modules may be required in each of the product variants. Variant-specific solutions need on the other hand not be reused.

Platform strategies appear to be less applicable for companies that can not afford the initial costs and risks involved, or can not predict future products with sufficient confidence to justify the investment. These companies are probably more interested in knowing what they can do incrementally that leads to predictable improvements in their product portfolio performance. Arguably, platform thinking may also be implemented bottom-up, by means of reusing design solutions to incrementally increase commonality while improving the variety to the market.

The design of the PFA requires a high level of technological and market knowledge, but the design of product variants can be relatively easy once the PFA has defined the design rules. Uncertainty about future technologies and demands is generally handled by preventive measures and by 'pushing' uncertainty-ridden design parameters to the variant-specific parts of the products, i.e. outside the reused platform subsystems – meaning uncertainty-ridden design solutions will probably deserve low levels of preparation for reuse.

The platform strategy literature contributes to the topic of this thesis in the following manner. It provides a framework to make the most of *predictable* reuse, by giving variety/commonality criteria to choose what to predetermine for reuse, and invest to maximise the benefits of that reuse. Furthermore, platform-thinking provides a visible separation of what is to be reused and what need not, facilitating reuse-related decision making for product variant design teams. So, what is missing? When future reuse needs are *not* predictable, the platform approach does not directly give advice of what to do. Furthermore, there is scarce support for incrementally developing a platform based on existing solutions and for renewing a platform continuously.

Lean product development

Lean product development focuses on achieving a flow of new developed products. Basically, this is done by continuous improvement of design solutions between product generations, yielding an 'incremental innovation' stream of products rather than radically innovative platforms followed by period of derivatives with low level of innovation. The lean approach does not appear to predetermine much for future reuse, but because design solutions are constantly improved, they become more suitable for reuse. Also, the lean approach stresses the importance of learning between projects. Therefore, reuse happens 'spontaneously' to a high degree, but alternatives to previous solutions are also explored. The lean approach requires rather highly skilled and motivated engineers, as much responsibility is put on them.

Lean approaches rely to a high degree on decision postponement for uncertainty handling. Uncertain design parameters are handled by considering entire ranges of parameter values in the designing until more information allows narrowing the specifications. This is called set-based design approach, and we consider it to be an important contribution to the topic of this thesis. Which issues are not directly dealt with by the lean product development approach? It can be argued that lean focuses on removing obstacles to its flow of new products, and not in finding radically new paths for these. So the lean approach may not be very applicable when a big-leap-forward is wanted with a radically new product introduced in a large project. In addition, lean approach literature does not seem to provide guidance on of how to invest for future reuse, meaning some kinds of reuse may be missed due to lack of development for reuse.

5 Results: Option-based reuse and continuous reuse management

In this chapter, the main contributions of this thesis are presented. We begin with a recapitulation of the previous chapters. Then we present the option-based reuse approach and finally our proposition for continuous reuse management.

5.1 Recapitulation of previous chapters

In the analysis chapters, we have identified different obstacles for ideal reuse practices, and corresponding areas of improvement potential. We can group these as into execution of reuse, and preparation for reuse.

To improve the execution of reuse, i.e. to capitalise as much as possible on existing solutions, companies should focus on

- Lower transfer costs through better design information tools and routines. Having searchable product design documentation systems that make it easy to capture design information 'on the fly' without significant effort, can greatly improve the product development efficiency, also regarding other aspects than design reuse.
- More accurate cost estimations as decision support. How much is the cost of introducing a new solution, vs. what is the cost of reusing an existing one? (This could be done by product areas, if solution-by-solution accounting is not practical.) Increasing the cost transparency of reuse decisions should improve their accuracy.
- Incentives and understanding of the potential benefits of reusing.

These improvement areas are vital for sound reuse practices, but since they are *conceptually* relatively straightforward, they are outside the main focus of this thesis.

To improve the *availability* of reusable solutions through development for reuse, companies should focus on:

- better forecasting of future reuse needs
- valuing of, and incentives/resources for *design for reuse*

It is this last point (the valuing of solution-specific investments in reusability) that this thesis aspires to support, as explained below.

What we have called *reuse approach*, is the manner in which preparation for reuse and actual reuse is bound together by a policy or plan. We have identified three conceptual types of reuse: ad-hoc, predetermined and option-based. Generally speaking, for a given design solution, the choice of reuse approach is mainly affected by two factors, namely

- Whether an *investment* is needed to enable reuse of the solution in future products
  - If it is necessary to adapt a design solution to be reusable, the preparation has a short-term cost that has to be budgeted.
  - If the design solution happens to be reusable without further design for reuse, then the choice of reuse approach for this solution is trivial (nothing needs to be done at this stage).
- Whether the future reuse needs are predictable

    - If the future reuse needs are predictable, the company can choose to make the investment in reusability and decide that the chosen solutions shall be reused in the future (predetermined reuse approach). Then the challenge is to quantify the net present value of the costs/benefits of the expected reuse, and to provide the resources for the investment.
    - If it is not possible to predict with confidence whether a given solution will turn out to be suitable for reuse, then any investments in its reusability should be based on an assessment whether the probability of reuse and benefits thereof outweighs the costs of preparation for reuse.

Hence, the difficult combination is when an investment in reusability is needed to enable reuse, but the company can not predict whether it will be appropriate to reuse the solution in the future. Unfortunately, this uncertainty often discourages investments in reusability, arguably partly because decision makers lack dedicated decision tools to deal with these issues explicitly. Therefore, we here propose an option-based valuing of development for reuse to contribute to sound reuse decisions under uncertainty.

How to justify the costs of specific instances of development for reuse? This would be trivial if the designing of reusable solutions happened *simultaneously* as their reuse, because then, the reusers could ‘pay a price’ for each reused solution that would amortise the solutions’ costs. The problem is that normally, when a solution is reused, the project that originally designed it has been dissolved. So if companies are to provide resources and incentives for development for reuse, they need to predict the probability of reuse of the prepared solutions, and value its reuse benefit. In short, there is a need for tools for *valuing of development for reuse*.

Such valuing of development for reuse should:

- Handle cases when reuse appropriateness is uncertain.
- Consider that more information becomes available during the process.
- Assist in scheduling the distribution of activities.
- Be able to guide individual instances of reuse so that its aggregate effect on the company is sound
- Be useful as a conceptual reasoning method, where factors are not easily quantifiable.

In the previous chapter we introduced a way of illustrating the flow of design solutions over product generations (see figure below). The vertical axis illustrates the level of ‘expected reusability’ of particular solutions at any given point in time. Solutions in the red area are one-off- or experimental solutions which are not immediately expected to be reused; solutions in the yellow area are seen as probably reusable; and solutions in the green area have been predetermined for reuse. By being proven by use and/or by being designed for reuse, some solutions will flow down in the figure to a higher level of reusability which will increase its chances of ‘surviving’ longer (moving to the right in the figure).

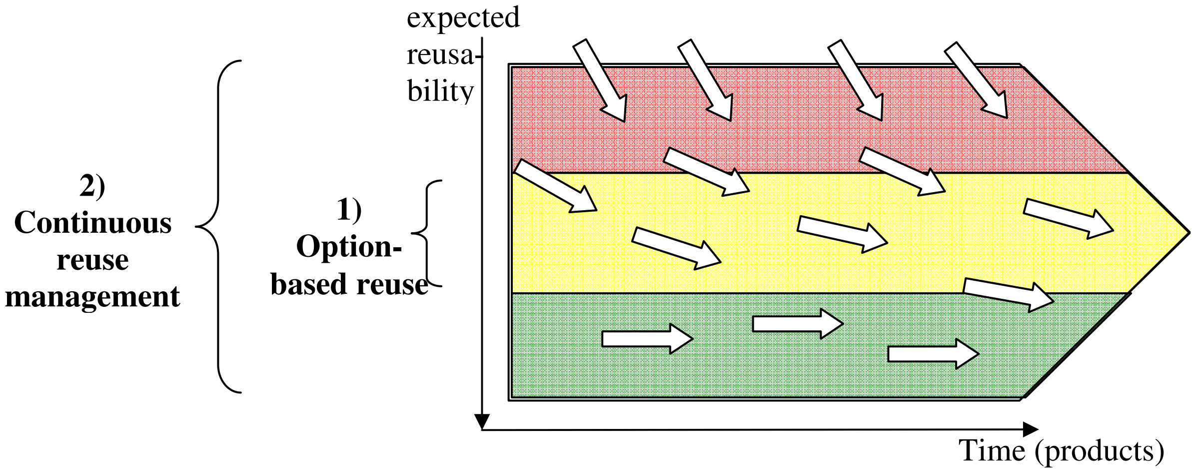


Figure 34: The two main contributions of this thesis.

This thesis has two main contributions, as indicated in the figure above:

1) An option-based reuse approach, which supports reuse decisions for design solutions in the yellow area, based on an uncertainty-aware real-option valuation of design for reuse.
2) A continuous reuse management framework, which covers how the flow of solutions should be steered across the three areas to achieve an optimum evolution of the product portfolio.

5.2 Introduction to real-option valuation

In this section, it is proposed that real option valuation can be used to value investments in design reuse. This is a tool whose underlying mindset is central in the reuse approaches that are presented in the subsequent sections.

Product development projects have a *"latent project value"* in addition to the 'official' deliverables [Ford and Sobek 2005]. An example of latent project value can be a contribution to organisational learning. In the same manner, many design solutions have a latent value that can be exploited through reuse. This is the value that is exploited when the solution is reused with a net benefit as a result, through variety cost reduction and/or reduced lead time. This latent value can often be increased through preparation for reuse. The problem is that the latent value often is difficult to discover and quantify, so it is often neglected. If the latent value would be obvious, companies would probably be much more prone to invest in reusable designs and amortise such investments through more efficient introduction of future products.

5.2.1 Real options in product development

When it is not decided beforehand whether a given design solution shall be reused, the value of its eventual preparation for reuse lies in the *added option to reuse*. 'Real option valuation' is a decision tool that can aid in selecting which solutions to prepare for future reuse. It is based on Option Pricing Theory which provides financial techniques [Faulkner 1996, Coldrick et al 2005]. It is suitable for product development which often has the characteristics of *"medium to long project lives, principal investment delayed to some point in the future and recognition of the fact that risk varies throughout the process"* [Doctor et al 2001 p.82].

Real option valuation is described as follows.

> *"When investing into a real asset, such as a development project (e.g., a product family design), an initial investment needs to be made (e.g., the platform design). Usually this is a small investment compared to the investment that will be made in the future to commercialize the resulting product (e.g., create the variants). A real option exists because the firm has the choice to drop the project and not make the commercialization investment if the development goes poorly or the market situation changes. In that case, the only loss incurred is the initial investment. On the other hand, if the expected return from the products looks good, the firm would likely make the additional investment (e.g., design the first variant). The option to make or decline additional investments into a design, for example, investing into the development of a new variant, has value to the company."* [Gonzalez-Zugasti 2001, p35]

Real option valuation appears to be useful to value the reusability of design solutions even as complex as product family architectures (PFA). For example, it can be used to support selection among architectural alternatives by valuing the *flexibility* that a given PFA provides to cope with future customisation requirements. Jiao et al propose a real option model in which *"product family design within a PFA is referred to as an investment strategy being crafted by a series of options that are continuously exercised to achieve expected returns on investment"* [Jiao et al 2006]. However, we note that once the PFA is introduced, usually its reuse by subsequent variant projects is mandatory, so there are no real options regarding its reuse. We argue that options-thinking are most useful to justify investments in reuse when the future projects will have real possibilities to choose whether to reuse or not.

5.3 An option-based reuse approach

In this section, we present a new approach to explicitly deal with reuse decisions in cases where creation and exploitation of *reuse potential* is significant.

> *"The central premise of real options is that, if future conditions are uncertain and changing the strategy later incurs substantial costs, then investing in flexible strategies can increase overall project value."* [Ford and Sobek 2005 p.175]

5.3.1 What is the option-based reuse approach?

Basically, the option-based reuse approach is

- valuing the reuse potential of design solutions across consecutive development projects,
- augmenting the reuse potential of selected design solutions through preparation for reuse,
- exploiting the reuse potential of past solutions by exercising the *option* to reuse if deemed suitable.

The approach focuses on maximising the *flexibility* for future projects to decide what they should reuse from the past in order to capitalise on the deliverables of previous projects. When successfully applied, the benefits of better reuse options outweigh the costs of preparation for reuse.

Key to this approach is that the *uncertainty* about present and future effects of reuse actions is incorporated into the concept of reuse potential, thus contributing more pragmatically to the needs of decision makers. Acknowledging this uncertainty provides a flexibility that can help companies adapt to unexpected product demands, and allow for smaller investments in reusability when appropriate.

In the predetermined reuse approach, the solutions to be reused must be technically functional in the future products to avoid unexpected disruption to the operations. Therefore, it is paramount that well known stable technologies are chosen for future reuse. By contrast, the option-based approach aims at providing potentially reusable solutions where the risk of technical unsuitability is part of the calculation, and therefore no unexpected disruption of plans needs to follow if the prepared solutions turn out to be unusable. Therefore, newer and less known solution technologies can be prepared for potential reuse. The investments for reuse are aimed at providing the best possible palette of options – i.e. maximising the reuse potential of selected design solutions by preparing them for reuse, and even alternative design solutions can be prepared for reuse, to maximise the options for future projects.

The option-based reuse approach is inherently bottom-up, in the sense that the reasoning can be applied at an arbitrary detail level. The option-based reuse approach may be implemented incrementally – it is not all-or-nothing, because the development for reuse can be based on existing designs or designs under development, and reuse practices can be improved gradually. Therefore, the approach is not dependent on large investments to be applicable. Since risk (probability of investments not leading to actual reuse) is incorporated into the concept of reuse potential, the approach allows for an explicit risk management.

The option-based reuse approach complements ad-hoc reuse and predetermined reuse. It is not limited to a set of products, but can in principle be used continuously over entire lifetimes of product families and even across several consecutive product families.

5.3.2 When is the option-based reuse approach appropriate?

We have seen that there are two conceptually different approaches to prepared reuse:

- a predetermined approach, where the reuse is planned for and decided beforehand, and
- an options-based approach, where preparation is done with the aim to maximise reusability without imposing it

Predetermined reuse is preferable when the reuse needs are predicted with confidence, while if the risk is considerable that the prepared solutions will prove to be unsuitable, it is better to postpone the decision to reuse until the moment of possible reuse.

In some cases, option-based reuse approach should replace the ad-hoc approach, to exploit the reuse potential of developed solutions better, i.e. becoming more aware about reuse without loosing flexibility. In other cases, option-based approach should replace predetermined reuse approach, because more flexibility is needed to adapt to changing technical circumstances in the future.

Option-based valuing of reuse investments is especially relevant to *solution-specific* decisions. Option-based valuing is not particularly useful to assess generic investments to increase reuse efficiency, since the effects of these are normally not meant to be optional. Generic investments in reusability may therefore be better to value by other means such as calculating the increased designer efficiency.

Option-based reuse compared to the set-based design

Ford and Sobek [2005] provide us with an example of real options in new product development. They model the flexibility value of pursuing different solution alternatives in parallel, the set-based approach, during one project.

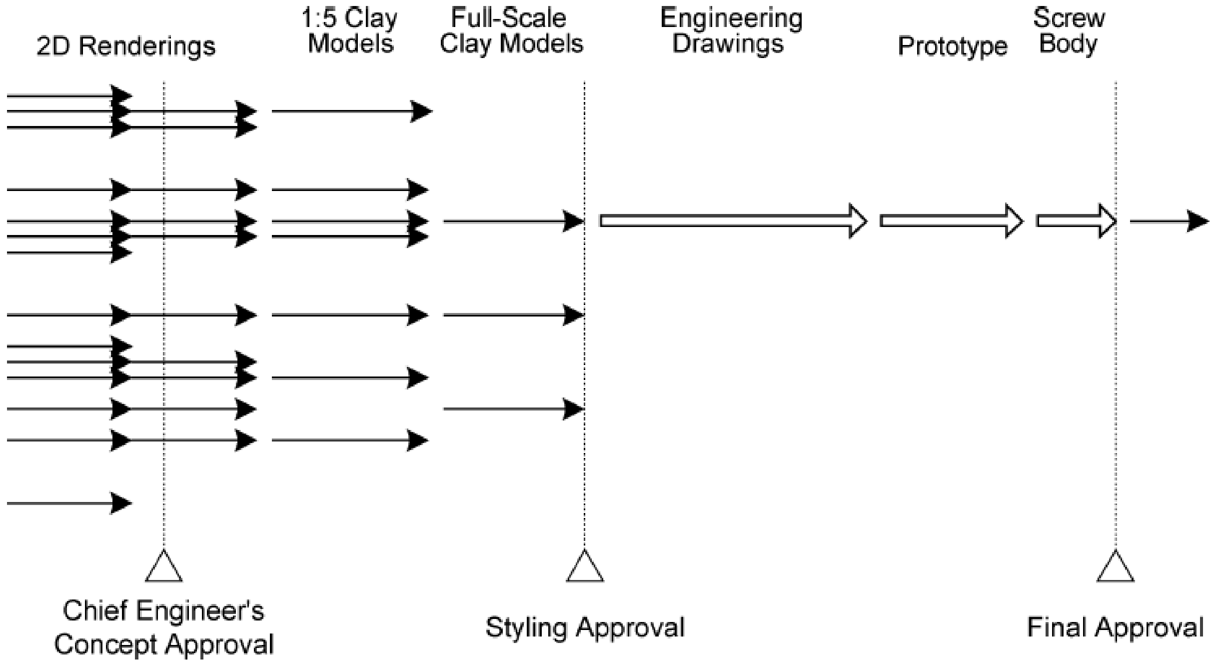


Figure 35: Set-based design at Toyota [Ford and Sobek 2005]

In the set-based approach, the extra effort spent on design alternatives which eventually are discarded is outweighed by the improved quality of alternative selection. In other words, when a solution is eventually chosen as the best of the considered set, this decision is based on better knowledge of the alternatives and is thus more likely to be accurate.

In relation to option-based design reuse, a similar reasoning can be applied. The main difference is that the creation of options is to be used *across* projects, and that new projects can choose from either reusing an existing solution or designing a new one.

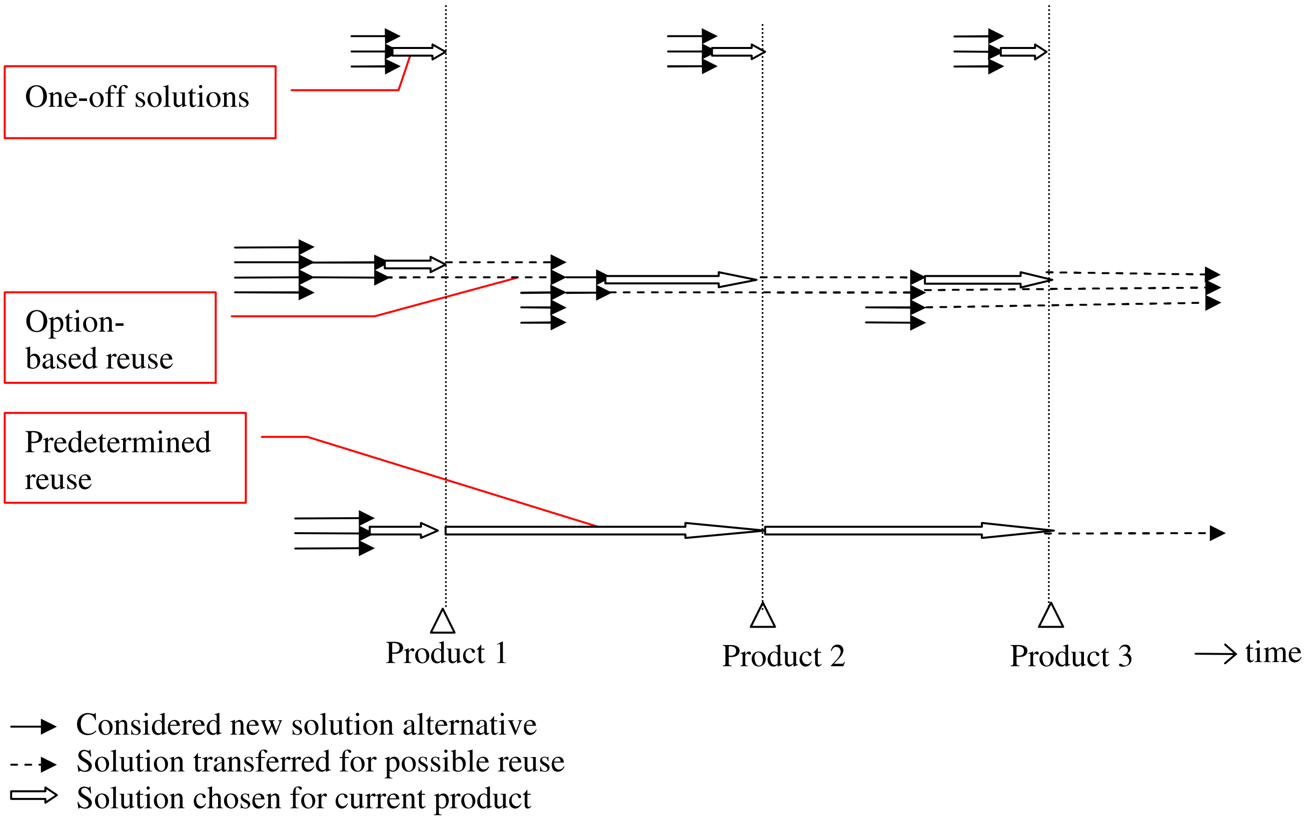


Figure 36: Option-based reuse in analogy to set-based design

5.4 Continuous reuse management

We propose that a continuous uncertainty-aware reuse management be implemented to steer reuse between evolving products, i.e. where there is great potential for reuse between product generations but the exact reuse needs are not predictable. This adaptive approach falls in line with the industry and research trend [Lehtonen 2005], and has similarities with Dynamic Modularisation [Riitahuhta 2001], in that reuse is managed proactively and reusable designs are developed and introduced 'continuously' to adapt to changing capabilities and environments.

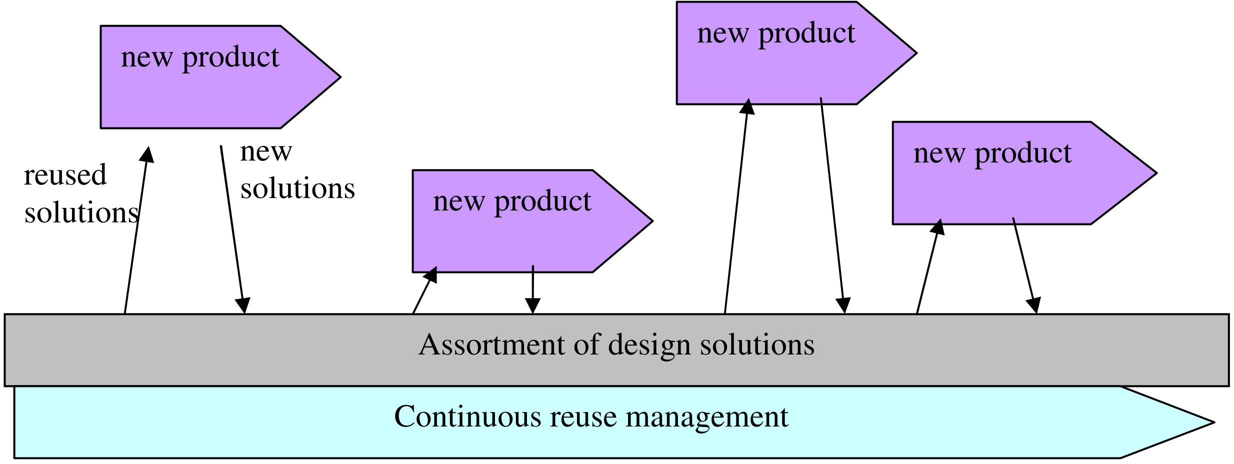


Figure 37: Continuous reuse management

5.4.1 Proposed duties

Concretely, we propose that reuse management should have the following duties over consecutive product development projects:

1) Monitor the reuse potential of past and new design solutions.
    a. Select which previous solutions to reuse in products under development.
    b. Select which new solutions to prepare for future reuse.
2) Allocate resources for investments in reuse
3) Identify where reuse decisions should be centralised and decentralised, respectively
    a. Create design rules to guide design reuse decisions, to allow for coherent decentralisation
4) Invest in improving general reuse efficiency through improved tools and routines
    a. identify how efficiency can be improved (attitudes? IT-tools? routines?)

Assessing the reuse potential of design solutions

A reuse strategy must select which design solutions the company should reuse over time based on business-strategic factors and the allocation of resources and risks over time. Solutions should be sorted according to reusability into

- one-off solutions (non-recurring functional requirements, or design solutions that are known to be unsatisfactory in the future)
- experimental solutions (behaviour of the solution is to be studied)
- reusable solutions (recurrent functional requirement, solution expected satisfactory)

The following factors should be considered

- the supply-chain costs of the design solutions
- the technological knowledge behind design solutions
- the product strategy, aligned with marketing plans
- the relative maturity, so that innovation efforts are spent where most needed

Determining when it is appropriate to reuse a past solution and when not, would be evident if the effects could be easily quantified. But since such calculations can be very cumbersome, an option is to group design solutions according to impact of *variety cost drivers*. This means identifying product areas of different reuse importance:

- Where does variety have the most value to the customer?
- Where does variety cost the most?
- Where may reuse reduce/increase product development costs/lead-time?
- Where may reuse reduce/increase production and logistics costs?

Such variety cost drivers may be derived from understanding the company-specific strategic weighting of aspects such as:

- PD lead time (gained/lost market share)
- PD costs
- production and logistics costs
- strategic impact (customer perception, core competence development)

For example, when deciding on whether to reuse a component or not, the production batch size may be one of the essential factors affecting the cost structure in high-volume products, because it determines how many units are to share the one-off development and tooling costs. In that case, reuse decisions should aim to minimise variable unit costs.

By contrast, in development-intensive products, the product development costs are large, so the reuse-related decisions should aim to minimise needed development effort and maximise the use of invested development efforts [Krishnan and Zhu 2006]. Especially, decision makers should consider whether *expertise* needed to (re)design a particular solution is available. Some solutions may demand considerable more design effort to redesign than to reuse, because the available expertise is not available in-house. On the other hand, if a solution is easy to replicate or redesign, it will not save much development resources to reuse it, so other cost factors such will have more relative importance.

Finally, reuse management should take into account that design reuse can impact the long-term competitiveness of companies, by enabling (or obstructing) the development of core capabilities in the company:

1) Reuse of strategically 'uninteresting' design solutions, to save development efforts, can free resources for experimentation and learning with more promising technologies.
2) Efforts in designing for reuse (e.g. designing a reuse-friendly product structure) can yield deeper understanding of the involved technologies, which can give a long-term competitive advantage to the company.

Allocating resources for preparation of solutions for reuse

How to justify investments in reusability? Real option-valuing is useful when it is unknown whether the solution will be adequate for reuse in future projects. Of course, any investment is based on an assumption that there is a significant *likeliness* that the solution will be reused, which depends on whether the design problem reoccurs and the solution is good enough, comparable to other possible competing solutions ('P' stands for 'probability'):

P(reuse adequate in future) = P(requirement reoccurring) * P(solution is satisfactory)

Furthermore, any investment is based on a belief that the potential reuse will bring benefits, which can be regarded as savings, to the future projects. These savings should come from the difference in development, supply-chain and variety costs of reusing the solution compared to introducing a new one.

Savings by reusing = (estimated savings in PD) + (estimated savings in supply chain)

These factors can then be combined in the present value of the reuse cycle (adapted from Baldwin and Clark 2000]:

Net Option Value = P(reuse adequate in future) * (savings by reuse)
– (cost of preparation for reuse) – (transfer cost) + (indirect impact)

where *indirect impact* include contributors that can be both positive and negative, e.g.

- product-strategic impact (is reuse in line with product strategy?)
- organisational cost/benefit of decision making (cost of planning, coordination, lead-time)
- learning effects

Postpone preparation for reuse?

When should a particular solution be prepared for reuse? Preparation for reuse need not be all or nothing, since often there is a possibility to complete the preparation for reuse at a later moment. But often it is cheaper to do the preparation for reuse at the same time as the reusable solution is originally designed. For example, the effort of thorough testing of one component may bring extra value to a future project if it reuses the component. In many cases there is also a minimum level of 'reusability' that has to be reached if the solution in the future is to be perceived as attractive for reuse at all.

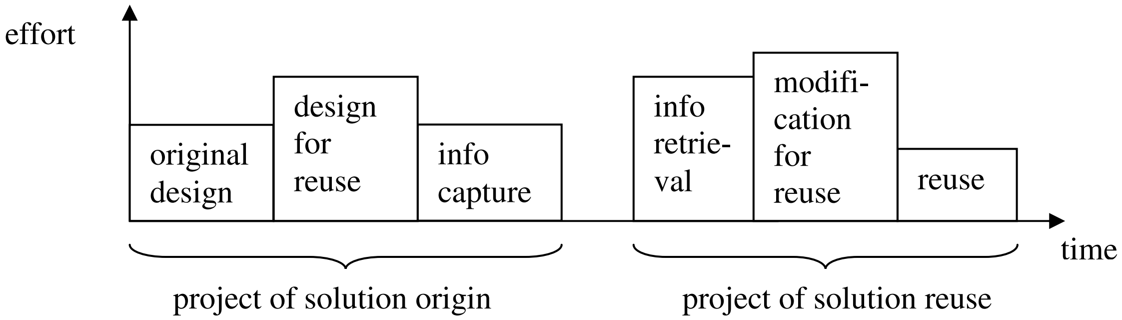


Figure 38: The effort costs of the reuse process are distributed over time (the effort levels in the figure are arbitrary for this example). Preparation for reuse may sometimes be postponed, at the cost of more required effort.

To a certain degree, a company can choose *when* to execute certain activities, see figure above. For instance, less effort to document the design solution in the original project can be compensated by more investigation in the later project (which often be at higher cost, of course). If, at the beginning of the timeline, the company is certain that the reuse will take place, then a Net Present Value analysis could guide when it is appropriate to schedule the efforts. On the other hand, if the future appropriateness of reusing the solution is uncertain, an real options analysis is more appropriate, because such analysis considers the possibility of skipping the later activities completely at any moment, if and when new information shows that reuse is inappropriate.

If it is very uncertain a particular solution will be useful for reuse in future products, it appears convenient to leave larger efforts for reusability to the later project (e.g. modify the solution for reuse, if needed), but it is vital that the original project *enables* reuse by certain minimum efforts, like documenting the solution well and not unnecessarily complicating the design.

For example: Consider that it requires X man/hours to test and document a design solution *specifically* for future reuse at the project of origin, and four times more, 4X man/hours to reconstruct the same information later at the project of reuse. Then this preparation for reuse in the originating project could be regarded as justified as long as there was a 25% chance of reuse (*assuming all man/hours of both projects are regarded equivalent and all other aspects disregarded*). If the likeliness of reuse is deemed lower than 25%, it could be justified to leave more work for the future project.

Allocation of different reuse-related decisions
Reuse management should identify which reuse-decisions that should be made outside projects, and which within. Which general rules that can be formulated, and which specific case-by-case decisions. This should be based on the strategic impact of decisions and need for coordination with other decisions.

Proposed activity list for reuse management
The following actions should be based on the 'reuse vision' for the company.

1) Prepare reuse plan
    a. Specify what to reuse from previous projects
    b. Forecast reuse opportunities in future projects
    c. Analyse and learn from reuse experiences from previous projects
    d. Classify reuse options according to importance
2) Keeping a reuse-plan up to date:
    a. Evaluate identified reuse options
    b. Classify probability of reuse options (low-medium-high)
    c. Re-evaluate reuse options periodically
3) Implement reuse plan
    a. Keep project management informed about actual reuse options, with expected cost reductions.
    b. Delegate responsibility for monitoring of different reuse options
    c. Identify and prepare for needed adjustments
    d. Update design reuse database

These actions should be adapted according to the current availability of resources for investments for reuse, which typically varies cyclically driving the product development organisation between being closer to 'fire-fighting mode' and being closer to 'long-term investment mode'. Furthermore, reuse management should strive to support the organisational learning from reuse, for example by having lessons-learned meetings and documents, considering successful and failed instances of design reuse.

5.5 Example

Consider this fictive case:

MobileMaker is a company that manufactures mobile phones. They target a niche in the market which demands a new phone to be launched each 3 months. Each new phone must be perceived as innovative, high quality, and trend-aligned to have a chance of maintaining the market share in the hard competition. This puts high pressure on the product development department.

We will follow three fictive reuse-related choices for new mobile phone versions. These regard three different features:

1. The processor
2. The display controller
3. The transceiver

For didactic reasons, we are only considering a few cost/benefit aspects when discussing these choices. In reality, many more factors would need to be considered.

The processor

Predictability of reuse opportunity

The processor solution is expected to be reusable as-is in the next generation of mobile phones. The processor was introduced one year earlier and has been proven fully satisfactory functionality-wise. There are no expectations of changed customer preferences affecting it.

Cost of variety, development
If a new processor solution is to be developed in the next generation of phones, it will need 10 times more development effort than reusing the current solution, which is a well-known off-the-shelf solution.

Cost of variety, production
If the processor production line cannot be reused, it will cost the processor supplier around x M$ in investment costs. If the reuse of the current processor model can be *guaranteed* for the next twelve months, the supplier offers a 40% discount on unit price (compared to the price of only ordering for the current phone model) due to the high volumes ordered.

Cost of development for reuse
Negligible, because the processor is a standard solution.

Extra production cost for reusable design
Negligible.

Conclusion
MobileMaker should choose **predetermined reuse** of the processor solution for the next 12 months. This means that the product development projects during this period should be required to reuse the processor. By choosing predetermined reuse of the processor, MobileMaker can order higher volumes and obtain better unit prices from the processor supplier.

The display controller

Predictability of reuse opportunity
The display controller solution was redesigned for the latest mobile phone that was launched, to meet increased graphics quality requirements. The customer preference trend is towards ever better display resolution and colour, driven by the possibility to take, send and receive pictures. The chosen technology for the display controller can be optimised more in future products with enough resources and knowledge. There are other technologies which may prove interesting in the future, but it is yet unknown which one the ‘winning’ technology will be in the longer term.

Cost of variety, development
The currently used display controller technology is rather complex, and a lot of design expertise has been invested in it. If the technology can be reused and fine-tuned, there will be significant savings in design and testing efforts, especially if a standard interface is used between the display controller and the processor. It would cost considerable (but not prohibitive) development effort and lead-time to develop a new solution from scratch. New display controller technology alternatives are being studied in the research laboratory.

Cost of variety, production
If the current display controller production facilities can be reused, the corresponding investments can be saved. But it is known that one of the drawbacks of the current display controller technology is that it is expensive to manufacture, which pushes its unit price high. The most promising prospective technology alternative (whose feasibility is yet unknown) is expected to have considerable lower variable manufacturing costs.

Cost of development for reuse
To enable partial reuse of the current display controller solution, the main design issue is the interface to the main processor device. If the interface can be standardised, then the display controller-processor combination can be altered without complex design coordination, which can mean considerable saving in development if the solution is reused in several product generations.

Standardising the interface would cost extra development resources which MobileMaker should provide to the project. Otherwise, it is expected that the future project will need to optimise some parameters of the display controller design anyway. The development team should carefully document how the technology works in general, and how they have proceeded to optimise it to the current application. Especially, the potential reusers will probably have great use of knowing the design rationale for different choices. The future project will have usage statistics, including failure rates, which they will want to compare with test protocols and experiences.

Extra production cost for reusable design
If the display controller-processor interface is standardised, the added connector will imply a 1,5% increase in variable unit cost for the current product.

Conclusion
Should MobileMaker go for a change of display controller technology in the next product? MobileMaker should consider how willing it is to take risks, and the strategic importance of quality and innovation. Probably, the best MobileMaker can do at this stage is to *postpone* the choice of display controller technology for the next phone generation, i.e. choosing an **option-based reuse approach**. MobileMaker should strive to maintain all interesting options open, by investing in the standard display controller-processor interface and preparing for design optimisation of the solution in case of reuse. In the future, the evolution of the technological alternatives should be monitored. If the current technology continues to be better, or as long as it is equivalent to other alternatives, it should be pursued. But MobileMaker should also prepare to switch over to another technology if and when needed.

The transceiver

Predictability of reuse opportunity
The current transceiver solution uses one very innovative concept, which has not yet been proven in the market, and it remains uncertain if it will be a successful concept or not.

Cost of variety, development
MobileMaker can relatively easily design new transceiver solutions by combining elements in a design library, so there is not much development effort to gain by reusing the current solution.

Cost of variety, production
The current transceiver is produced by a supplier with which MobileMaker has no commitments beyond the current product. So it will not have any extra cost to switch to another transceiver solution in the future.

Cost of development for reuse
If the current solution would be prepared for reuse in several product generations, it should be thoroughly enhanced with regard to robustness, modularised and fitted with standard interfaces. This would cost a considerable amount of development resources.

Extra production cost for reusable design
If the transceiver was to be modularised, it would require investments (fixed costs) in a new production line at the supplier. These investments would significantly increase the unit cost if the solution was only used in the current product, but if the solution is reused in several future product generations, then the fixed costs would be spread out over such high volumes that they would be close to negligible.

Conclusion

Since it appears that it is very uncertain if the solution will be successful, and investments in reusability would be significant, MobileMaker should not invest in reusability of the current transceiver solution (**ad-hoc reuse approach**). MobileMaker should of course follow-up its performance in the market, and, only if deemed successful, improve and reuse it in future products.

Table 3: Example showing need for three different reuse approaches within same product series

Factor	**Processor**	**Display controller**	**Transceiver**	*Comment*
Expected reusability	High (as-is)	Probable (needs optimisation)	Uncertain	*Predicted fit between current solution and future needs*
Cost of variety	High	Medium, uncertain	Low	*Equivalent to 'cost of not reusing' or 'benefits of reusing'*
Cost of development for reuse	Low	Medium	High	*Effort needed to design for reuse, over-design cost*
Recommended reuse approach	***Predetermined***	***Option-based***	***Ad-hoc***	

6 Conclusion

In this last chapter, the main contributions from the thesis and the research method are discussed.

6.1 Summary of results

Our main research question was:

How can we model the main conceptual approaches to design reuse?
More specifically, we wanted to develop a model that dealt with uncertainty, drifting contexts and distribution over time of actions and effects.

We modelled the design reuse process as consisting of two phases, a preparation phase and an execution phase, which normally correspond to different projects separated in time. This allowed us to classify reuse approaches conceptually according to how deliberately these two phases are linked together, into ad-hoc, option-based, and predetermined reuse.

This thesis has proposed two conceptual tools with the ultimate goal of improving the reuse practices of companies, namely:

- An option-based reuse approach focusing on how to reason when making reuse decisions regarding design solutions whose future reusability is probable but not certain. Basically, it is about thinking in terms of the reuse potential of solutions, how to enhance and exploit it, and realising the option value of flexibility.
- A framework for continuous reuse management. This is based on a flow-oriented view of the assortment of design solutions in a company over time. The purpose of continuous reuse management is to steer the evolution of the product assortment by enhancing the reusability of selected solutions to increase their chances of surviving longer.

Our working questions were answered as follows:

- *What is good design reuse, i.e. what is the vision regarding design reuse?*
  We formulate good design reuse as the optimal reuse choices as could be seen in retrospect, with all facts at hand, including consideration that the companies' capabilities and its business environment evolve with time.

- *Where is there most improvement potential for companies?*
  We have identified a large number of issues where there might be potential for improvement, which are very company-dependent. At the highest level, each company should identify which of the three areas is of most need of improvement: 1) willingness to reuse; 2) availability of solutions suitable for reuse; and 3) means to transfer design solutions for reuse.

- *Which are the essential design reuse decisions?*
  When designing a potentially reusable solution, the two main decisions that shape the subsequent reuse process are 1) whether to develop for reuse; and 2) whether to predetermine its future reuse. In subsequent projects, the decision regards 3) whether to

reuse the solution (provided it is optional). These decisions can be actively or passively made, and are tied to requirements management. For reuse opportunities identified prior to a development project, explicit requirements for reuse can be issued, while for reuse opportunities identified during development projects, decisions have to be made 'on-the-fly', usually following formal or informal incentive systems.

- *How can the evolution of assortments of products and design solutions be modelled to visualise reuse patterns?*
  We have presented a new type of illustration that shows the 'flow' of design solutions across product generations, with time on the horizontal axis, and expected reusability in the vertical axis.

In addition, this thesis provides insights into the phenomenon of design reuse by its analysis from the six different viewpoints in chapter 3. By showing the multiplicity of mechanisms involved in design reuse, the thesis contributes to a more nuanced and holistic understanding of the phenomenon, discouraging beliefs that simple 'silver bullet' solutions can be found.

6.2 Evaluation

Since this study is of exploratory nature, the results are not 'verified' as true or false. Instead, the results are in this section validated as discussed in chapter 2, trying to make a convincing case for the usefulness of the results. This is done by discussing whether the results are

- *'correct'* (internally consistent and based on acceptable knowledge and assumptions) and
- *relevant* (addresses stated problems, is original complementing previous research)

6.2.1 Correctness

Here, we discuss whether the conclusions drawn are based on acceptable knowledge and following a logically correct reasoning.

Knowledge base and assumptions
The knowledge base for this study consists of

- an extensive literature study covering over 250 academic papers and books, of which about the 50% most interesting was used
- a set of interviews in Norwegian industry
- the author's own experiences from 4 years working as a software product developer at Enea OSE, a Realtime Operating Systems provider
- input from peers in academia and industry.

The literature study was, with some exceptions, restricted to literature having *product development* as a common denominator. This restriction, employed for practical reasons, may be a weakness of the study, since there is a vast knowledge base that doubtless would be relevant for our topic but does not specifically target product development. However, we argue that the study is reasonably balanced because of the variety of viewpoints covered, including: product market strategy, modelling of technical systems, project management, uncertainty handling, IT design and product management tools, supply chain design, financial tools for R&D, and others.

The delimitations and assumptions stated and discussed in chapter 2 were

- only high-tech manufactured products, so we assume the engineering costs and the supply chain costs are significant

- business networks not treated specifically, so we assume the company has control over its product portfolio
- too much or inappropriate reuse not treated
- branding-driven reuse not treated, and customer-perceived quality of product treated as given, so we assume main driver for reuse is costs and lead time
- details of *how to* design for reuse and design by reuse not treated
- ownership of design solutions not treated

With these delimitations and assumptions, we see that this study is not entirely relevant for companies manufacturing commodity low-tech products where development costs are negligible in comparison to manufacturing costs, or companies where the default behaviour is to reuse everything from previous products, or companies where reuse is mainly driven by branding and fashion.

Internal consistency
Does this study make a consistent use of concepts? The most important concepts used in this study are

- design solution
- assortment of design solutions and product assortment/portfolio
- design reuse, as-is reuse and partial reuse
- the reuse process and its phases
    a. preparation for reuse
    b. transfer of solutions
    c. reuse
- the individuals/teams executing the reuse process phases, i.e. the ones that
    a. the decide on and prepare a solution for reuse
    b. decide and reuse the solution
- the development project scope of the reuse process phases
    a. the project of solution origin
    b. the project of solution reuse

As has been discussed previously, we have deliberately left these concepts at an abstract level, often leaving it open for different concretisations. This regards specially the concept of ‘design solution’ and the boundaries of the phases of the design reuse process, leaving open questions like “what is a design solution?” and “where is the limit between preparation for reuse and transfer for reuse?” This ambiguity may be seen both as a strength and a weakness of this study. It is a strength because it is left open for situation-specific interpretations, and a weakness because it may result unclear for the reader concretely what is meant in different statements. We have tried to bridge this gap by providing many examples complementing the line of reasoning.

The decision making has been modelled as consisting of discrete decisions, albeit we have recognised these can be active or passive. However, in reality, and unfortunately for us, decisions are often made ‘gradually’, for example by discussing and trying out a proposition until objections are weak enough to proceed, and thus it may sometimes be difficult to claim that a particular decision has been made at a particular point in time, as we have modelled. This especially concerns our classification of reuse approaches between ad-hoc, option-based and predetermined, which to be more nuanced would have to consist of a continuous scale ranging from ‘no future reuse expected’ to ‘future reuse firmly expected’. This is actually the way it is represented in our proposed flow-oriented illustration of the lifecycles of design solutions. We see no contradiction in principle between these two modelling approaches.

Furthermore, as we have discussed before, design reuse may take many forms and hence our modelling of the decision whether to reuse or not may be too categorical, and only applicable to discrete component reuse. At the ‘softest’ extreme, design reuse may regard general design knowledge, such as lessons learned from failed solutions. If this was to be considered, we would have to model the decision of ‘how much to reuse if anything’. However, we argue that the categorisations made are reasonable for the sake of modelling the conceptual approach alternatives.

6.2.2 Relevance

The relevance of this study as a research contribution can be discussed in these steps:

1. Does the stated research motivation address a ‘real’ need?
2. Are the research questions good for the stated research motivation?
3. Do the results answer the research questions?
4. Do the results *complement* previous research, i.e. is it original and based on accepted knowledge or explicitly challenging it?

The research motivation was the perceived problem of missed reuse opportunities, and the need for gradual approaches to improve the evolution of product portfolios over time by means of reuse. The rationale for this choice of research motivation is discussed in the introduction to this thesis, chapter 1. The rationale for the choice of research questions is discussed in chapter 2, and the answers to the research question and working questions is summarised above. Below, we discuss the potential usefulness of the results for academia and industry.

Contribution to the research community

The perceived knowledge gap, as previously discussed, regards decision making around design reuse over time and under uncertainty. While there are many contributions to optimisation of design reuse in different forms, most of these rely on *known* optimisation criteria and solution spaces, and have shortcomings dealing with real-life situations which often are ridden with uncertainty.

This study contributes to fill this gap with an uncertainty-aware framework for decision-making, which we have called continuous reuse management. The ‘uncertainty-awareness’ resides in that the framework supports humans’ reasoning upon uncertainty by using ‘soft’ and subjective dimensions such as ‘expected reusability’, ‘confidence’, etc, and focusing on the conceptually different decision alternatives, such as “decide now or postpone decision?”, as opposed to models for ‘automated’ decision-making relying on equations and hard data. We believe that companies that are successful in their reuse practices apply a pattern of decisions that can be described by the here proposed framework.

The continuous reuse management framework combines elements from many ‘bodies of knowledge’, such as project management, portfolio management, engineering design methodology, decision support tools, financial tools, etc. We regard it as a rather pragmatic approach that encourages a more nuanced thinking about the requisites to achieve good design reuse, and the effects thereof.

The originality of the results can be found in different areas; at least as far as our literature study has shown. First, the modelling of the design reuse process explicitly following the lifecycles of design solutions across several products in time is novel. Second, the application of real options valuing (stemming from financial theory) specifically to the *design reuse process* has not been done in previous research, however many previous contributions have been very close, e.g. by using real options to value product platforms. Third, the concept of a continuous reuse management, incorporating projected reuse expectancy, reactive and proactive actions, at both micro- and macro-level (individual design solutions and evolution of the entire solution assortment, respectively) into a flow-inspired model, is to our knowledge original.

These results complement product platform-based approaches by incorporating explicit uncertainty awareness in the decision making. The results borrow the flow-thinking of lean design and complement it by explicitly supporting valuing of solution-specific investments in reusability.

The progression of this PhD study has been reflected in conference papers, which have been presented at engineering design conferences (listed in the section on the research method) and thereby have been challenged by representatives from the research community. This, together with in-depth discussions with peer researchers at these conferences and at different universities, can be said to indicate a degree of acceptance by the research community.

Industrial contribution

Are the results of this thesis usable by industry?

We believe that the result of our thesis in reality is a formalisation of what in successful companies could be called ‘common sense’. The main contribution is therefore to provide conceptual tools that can help this become *explicit*, and increase the awareness on the underlying mechanisms of design reuse. Naturally, the thesis follows an academic tradition that is per se not the best format to be easily accessible in industry. This regards for example the discussions about different uses of terms and concepts. Despite this, we argue that this thesis contributes to industry in the following manner.

First, the thesis provides a broad knowledge base on the dynamics of design reuse which can be used by companies to ‘know themselves’ better in order to tailor reuse-related strategies to their specific needs. This contribution is ‘balanced’ in the sense that it highlights both pro’s and con’s of reuse. It is expected to be usable for diagnosing causes of long-term inefficiencies in product development. Second, the proposed option-based reuse approach provides concepts and terms (such as ‘reuse potential’, and ‘option to reuse’ and ‘option value of preparation for reuse’) that we argue are usable to facilitate discussions towards decision making on reuse under uncertainty. Third, we hope that the continuous reuse management framework can be incorporated in decision forums at companies, both providing a ‘check-list’ of reuse-related issues that should be thought about, and means for verbal and visual articulation of these issues. We see continuous reuse management not as an autonomous decision process, but one that is integrated into portfolio management and detail-level engineering decision making. We do not get into the discussion of how exactly to allocate the responsibility for continuous reuse management, but hope companies can tailor solutions according to their needs.

The problem of missed reuse opportunities is specifically addressed by this thesis as follows. The missed reuse opportunities due to lack of understanding of the reuse benefits are countered by the insights offered on the economic effects of reuse. The missed reuse opportunities due to lack of efficient knowledge transfer tools are countered by the investigation of transfer needs and the presentation of available tools. The missed reuse opportunities due to lack of available solutions adequate for reuse are countered by the presentation of a flow oriented view on solution development which should encourage designing for reuse, and thus gradually enrich the reusable content of the company's assortment of solutions.

6.2.3 Evaluation of research method

This research is based on an initial empirical research mapping the problem area, and an extensive literature review used to develop the two new conceptual tools: the option-based reuse approach and the continuous reuse management. Much of the reasoning used to arrive at the results from the literature base has been abductive reasoning, i.e. proposing likely explanations for the available observations, which is to say that the used research method is mainly *exploratory*.

We have focused our 'exploration' to develop a framework combining elements from different knowledge fields aiming for a balanced and generic modelling of the design reuse phenomenon in a form that is useful as decision support. Therefore, we have not prioritised verification of previous and new tools by means of empirical studies. Arguably, 'verification' of the proposed tools by empirical means could prove practically nearly impossible as it would require tracing the effects of the many factors, internal and external, that affect the performance of product portfolios over time, in order to know the true effect of different reuse-related decisions. On the other hand, it is regrettable that time limits did not allow us to execute case studies to validate how well the proposed framework can be used to *describe* the reuse-related decision making that goes on at companies.

Research in engineering design is special since it has to deal both with exact sciences (physical laws, mathematics) of artefacts and the social sciences of the humans designing, using or otherwise involved with artefacts. We have tried to maintain a reasonable balance between the different research traditions that these represent. However, being from an engineering schooling, we have to acknowledge limitations in knowledge about non-engineering domains, especially regarding organisational behaviour. Our selection of literature has also been 'biased' towards research coming from the 'engineering community', which unfortunately too often appear to apply an 'exact science'-approach to studying human and organisational behaviour.

However, considering the available time for this one PhD research, we consider that the chosen research method has been satisfactory and has lead to a significant contribution to both the research community and industry.

The presentation of the results in this thesis is hoped to be clear enough and accessible not only for research peers but for representatives from industry. Arguably, a piece of research is of less validity if it is not communicable. Therefore, efforts have been made to illustrate abstract discussions with concrete examples and figures. In addition, both peers with field-specific background and peers with other backgrounds have been engaged to read drafts of this thesis and provide feedback on its readability.

6.3 Suggestions for future research

We have classified the causes of missed reuse opportunities into lack of willingness, availability of reusable solutions, and means for design transfer. However, we have not quantified the relative importance of these categories, which we deem would be very interesting to know. One beginning could be to let a number of companies answer questionnaire grading their perception of causes of missed reuse.

It is believed that the essence of continuous reuse management is actually common practice at many successful companies, only under other names or implicitly. Therefore it would be interesting to make case studies exploring whether actual practices at companies can be *described* using the results of this thesis. Such case studies could observe decision-making in action and investigate why particular solutions have been reused, and why others have not. In addition, it would be interesting to test whether the reuse framework presented here can contribute to make the reuse process more visible, thus encouraging its improvement.

In the area of design for reuse, we have found that often the challenge is the uncertainty about what the capabilities and needs will be in the future. In such cases, designers need to 'design for uncertainty', which we suggest that future research could explore, as has been done with other 'design for X' methodologies.

Another area of interest that is important to understand and improve the reuse behaviour of companies, is the issue of ownership of design solutions. This issue had to be excluded from this study because of time limits. The allocation of responsibility for different solutions (or solution levels/types) appears to affect the possibilities to specialise knowledge and ensure coordination of decisions. In addition, it seems plausible that the identification that a designer experiences with a solution he/she has contributed to design has considerable impact on the motivation to prepare, document, change, and reuse it, thus contributing to an 'irrational' element in decisions.

Finally, this thesis has found that product platform methodologies often are excessively simplistic and rigid, which does not fit well in the ambiguous and changing world of many companies. Therefore, we suggest that future research could aim at finding more flexible and gradual formulations of platform strategies using the concepts presented in this thesis.

7 References

Ahonen, T & Nurmi, J 2004
    ‘Design reuse and design for reuse, a case study on HDSL2’, Proceedings of International Symposium on System-on-Chip, IEEE, p 129-133
Andreasen, MM, McAloone, T & Mortensen, NH 2001
    P* Insight report on Multi-product development – platforms and modularisation, Dept. of Mechanical Engineering, Technical University of Denmark, Lyngby
Baldwin, C & Clark, K 1997
    ‘Managing in the age of modularity’, Harvard Business Review, Sept/Oct, p 84-93
Baldwin, C & Clark, K 2000
    Design Rules: The Power of Modularity, The MIT Press, Massachusetts
Ball, L, Lambell, N, Ormerod, T, Slavin, S & Mariani, J 2001
    ’Representing design rationale to support innovative design reuse - a minimalist approach’, Automation in Construction, v 10, p 663–674
Barnes, BH & Bollinger, TB 1991
    ‘Making reuse cost-effective’, IEEE Software, Jan, p 13-24
Belecheanu, R, Pawar, KS, Barson, RJ, Bredehorst, B & Weber, F 2003
    ‘The application of case based reasoning to decision support in new product development’, Integrated Manufacturing Systems, v 14, n 1, p 36-45
Berglund, F & Claesson, A 2005
    ‘Utilising the concept of a design’s bandwidth to achieve product platform effectiveness’, 15th International Conference on Engineering Design ICED05, Melbourne
Blackenfelt, M 2001
    ‘Managing complexity by product modularisation: Balancing the aspects of technology and business during the design process’, Doctoral Thesis, Royal Institute of Technology, Stockholm
Blessing, L 2005
    ‘A Design Research Methodology’, Lecture notes, The Design Society’s 7th Summer School on Engineering Design Research
Bohn, R 2000
    ‘Stop fighting fires’, Harvard Business Review, July-August, p 83-91
Busby, JS 1998
    ‘Effective practices in design transfer’, Research in Engineering Design, v 10, p 178-188
Busby, JS 1999
    ‘The problem with design reuse: an investigation into outcomes and antecedents’, Journal of Engineering Design, v 10, n 3, p 277-296
Claesson, A 2006
    ‘A configurable component framework supporting platform-based product development’, PhD thesis, Chalmers University of Technology, Göteborg
Clark, KB & Fujimoto, T 1991
    Product Development Performance, Harvard Business School Press, Cambridge
Chen, KM & Liu, RJ 2005
    ‘Interface strategies in modular product innovation’, Technovation, v 25, p 771-782
Coldrick, S, Longhurst, P, Iveya, P & Hannis, J 2005

'An R&D options selection model for investment decisions', Technovation, v 25, p185–193

Cooper, R 2001
Winning at New Products – Accelerating the process from idea to launch, Third edition, Basic Books, New York

Crossland, R, Williams, JHS & McMahon, CA 2003
'An object-oriented modeling framework for representing uncertainty in early variant design', Research in Engineering Design, v 14, p 173-183

D'Souza, B & Simpson, T 2003
'A genetic algorithm based method for product family design optimisation', Engineering Optimization, v 35, n 1, p 1-18

Dalton, JA, Norman, PW, Sen, P & Whittle, S 2005
'Extending the design opportunities afforded by modelling frameworks', 15th International Conference on Engineering Design ICED05, Melbourne

Dana, JD 2003
'Remark on 'Appropriateness and impact of platform-based product development'', Management Science, v 49, n 9, p 1264-1267

Davies, A & Brady, T 2000
'Organizational capabilities and learning in complex product systems: Towards repeatable solutions', Research Policy, v 29, p 931-953

Desai, P, Kekre, S, Radhakrishnan, S & Srinivasan, K 2001
'Product Differentiation and Commonality in Design: Balancing Revenue and Cost Drivers', Management Science, v 47, n 1, p 37–51

Doctor, RN, Newton, DP & Pearson, A 2001
'Managing uncertainty in research and development', Technovation v 21, p 79-90

Duffy AHB & Ferns AF 1999
'An analysis of design reuse benefits', in Proceedings of the International Conference on Engineering Design ICED99, Munich, v 2, p 799-804

Eckert, C, Clarkson, PJ & Zanker, W 2004
'Change and customization in complex engineering domains', Research in Engineering Design, v 15, p 1-21

Ericsson, A & Erixon, G 1999
Controlling Design Variants – Modular Product Platforms, Society of Manufacturing Engineers, Michigan

Faulkner T 1996
'Applying options thinking to R&D valuation', Research Technology Management, v 39, n 3, p 50-56

Fine, C 1998
Clockspeed – Winning industry control in the age of temporary advantage, Basic Books, New York

Fisher, M, Ramdas, K & Ulrich, K 1999
'Component Sharing in the Management of Product Variety', Management Science, v 45, n 3, p 297-315

Fiore, C 2005
Accelerated Product Development – Combining Lean and Six Sigma for peak performance, Productivity Press, New York

Fixson, S 2004
'Product architecture assessment: a tool to link product, process and supply chain design decisions', Journal of Operations Management, v 23, p 345-369

Ford, D & Sobek II, D 2005
‘Adapting real options to new product development by modeling the Second Toyota Paradox’, IEEE Transactions on Engineering Management, v 52, n 2, p 175-185
Fujita, K 2006
‘Product variety optimization - Simultaneous optimization of module combination and module attributes’, in Simpson, Siddique, Jiao (eds.) Product Platform and Product Family Design, Springer, New York
Gann, DM & Salter, A 2000
‘Innovation in project-based, service-enhanced firms: The construction of complex products and systems’, Research Policy, v 29, p 955-972
Garud, R & Kumaraswamy, A 1995
‘Technological and Organizational Designs for Realizing Economies of Substitution’, Strategic Management Journal, Special Issue, Summer, v 16, p 93-109
Gonzalez-Zugasti, J, Otto, K & Baker, J 2001
‘Assessing value in platformed product family design’, Research in Engineering Design, v 13, n 1, p 30-41
Gupta, AK & Wilemon, D 1996
‘Changing Patterns in Industrial R&D Management’, Journal of Product Innovation Management, v 13, n 6, p 497-511
Halman, J, Hofer, A & Van Vuuren, W 2003
‘Platform-driven development of product families: Linking theory with practice’, Journal of Product Innovation Management, v 20, p 149-162
Harlou, U 2006
‘Contribution to a theory of product families’, PhD Thesis, Technical University of Denmark, Lyngby
Hatchuel, A & Weil, B 2003
‘A new approach of innovative design: An introduction to C-K theory’, 14th International Conference on Engineering Design ICED03, Stockholm
Hauser, J 2001
‘Metrics thermostat’, The Journal of Product Innovation Management, v 18, p 134-153
Henderson, R & Clark, K 1990
‘Architectural innovation: The reconfiguration of existing product technologies and the failure of established firms’, Administrative Science Quarterly, v 35, n 1, p 9-30
Huang, J & Newell, S 2003
‘Knowledge integration processes and dynamics within the context of cross-functional projects’, International Journal of Project Management, v 21 p 167-176
Hubka, V & Eder, E 1988
Theory of Technical Systems, Springer-Verlag, Berlin
Ishii, K & Yang, T 2003
‘Modularity: International industry benchmarking and research roadmap’, Proceedings of ASME DETC’03, Chicago
Jensen, T, Nilsson, C 2005
‘Designing product families - Experiences of four Norwegian manufacturing companies’, 15th International Conference on Engineering Design ICED05, Melbourne
Jiang, L & Allada, V 2001
‘Robust modular product family design’, Intelligent Systems in Design and Manufacturing IV, Gunasekaran, A & Gopalakrishnan, B, editors, Proceedings of SPIE Vol. 4565
Jiao, J, Ma, Q & Tseng, M 2003

‘Towards high value-added products and services: mass customization and beyond’ Technovation, v 23, p 809–821

Jiao, J, Kumar, A & Lim, CM 2006
‘Flexibility valuation of product family architecture: a real-option approach’, International Journal of Advanced Manufacturing Technology, v 30, p 1-9

Jiao, J & Tseng, M 2000
‘Understanding product family for mass customisation by developing commonality indices’, Journal of Engineering Design, v 11, n 3, p 225-243

Johannesson, H & Claesson, A 2005
‘Systematic product platform design: a combined function-means and parametric modeling approach’, Journal of Engineering Design, v 16, n 1, p 25–43

Juuti, T, Lehtonen, T, Pulkkinen, A & Riitahuhta, A 2004
‘Efficient platform utilisation with configurable products – Enabling factors illustrated’, Proceedings of the 7th Workshop on product structuring – Product platform development, Göteborg

Juuti, T, Lehtonen, T 2006
‘Using multiple modular structures in delivering complex products’, Proceedings of the NordDesign 2006 conference, Reykjavik

Kappel, T 2001
‘Perspectives on road maps: how organizations talk about the future’, The Journal of Product Innovation Management, v 18, p 39-50

Kennedy, M 2004
‘The Toyota product development system’, Machine Design, v 76, n 9, p 152

Khurana, A & Rosenthal, R 1998
‘Towards holistic front ends in new product development’, Journal of Product Innovation Management, v 15 p 57-74

Kim, B 2003
‘Managing the transition of technology life cycle’, Technovation, v 23, p 371–381

Kim, S, Bracewell, R & Wallace, K 2005
‘A framework for design rationale retrieval’, 15th International Conference on Engineering Design ICED05, Melbourne

Kreimer, K 1995
‘In Search for Relevance: Project Management in Drifting Environments’, Scandinavian Journal of Management, v 11, n 4, p 335-346

Krishnan, V & Bhattacharya, S 2002
‘Technology selection and commitment in new product development: The role of uncertainty and design flexibility’, Management Science, v 48, n 3, p 313-327

Krishnan, V & Gupta, S 2001
‘Appropriateness and impact of platform-based product development’, Management Science, v 47, n 1, p 52-68

Krishnan, V & Ulrich, K 2001
‘Product development decisions: A review of the literature’, Management Science, v 47, n 1, p 1-21

Krishnan, V & Zhu, W 2006
‘Designing a family of development-intensive products’, Management Science, v 52, n 6, p 813-825

Kristjansson, A & Hildre, HP 2004

‘A framework for evaluating platforms in product developing organisations’, 7th Workshop on Product Structuring – Product Platform Development, Chalmers University of Technology, Göteborg

Lees, B, Hamza, M & Irgens, C 2000
‘Case-based reasoning support for engineering design’, Intelligent Systems in Design and Manufacturing III, Gopalakrishnan, B & Gunasekaran, A, editors, Proceedings of SPIE v 4192

Lehtonen, T, Juuti, T, Pulkkinen, A, Roikola, AM & Riitahuhta, A 2005
‘Origins and prospects of platform based product paradigm’, 15th International Conference on Engineering Design ICED05, Melbourne

Lehtonen, T, Juuti, T, Pulkkinen, A, Riithahuhta, A 2003
‘Dynamic Modularisation – A challenge for design process and product architecture’, 14th International Conference on Engineering Design ICED03, Stockholm

Li, Z, Anderson, D & Ramani, K 2005
‘Ontology-based design knowledge modeling for product retrieval’, 15th International Conference on Engineering Design ICED05, Melbourne

Lim, W 1994
‘Effects of reuse on quality, productivity and economics’, IEEE Software, September, p 23-30

Lim, W 1996
‘Reuse Economics: A comparison of Seventeen Models and Directions for Future Research’, IEEE International Conference on Software Reuse

Maier, J & Fadel, G 2001
‘Strategic decisions in the early stages of product family design’, ASME Design Engineering Technical Conference DETC’01, Pittsburg

Malik, K 2002
‘Aiding the technology manager: a conceptual model for intra-firm technology transfer’, Technovation, v 22, p 427–436

Martin, M & Ishii, K 1996
‘Design for variety: A methodology for understanding the costs of product proliferation’, ASME Design Engineering Technical Conferences 96-DETC, Irvine

Martin, M & Ishii, K 1997
‘Design for variety: Development of complexity indices and design charts’, Proceedings of ASME Design Technical Engineering Conferences DETC’97, Sacramento

Martin, M & Ishii, K 2002
‘Design for variety: developing standardized and modularised product platform architectures’, Research in Engineering Design, v 13, p 213-235

McAdams, D, Stone, R & Wood, K 1999
‘Functional Interdependence and Product Similarity Based on Customer Needs’, Research in Engineering Design, v 11, p 1–19

Messac, A, Martinez, M & Simpson, T 2002
‘Introduction of a product family penalty function using physical programming’, ASME Journal of Mechanical Design, v 124, n 2, p 164-172

Meyer, MH, Tertzakian, P & Utterback, J 1997
‘Metrics for managing research and development in the context of the product family’, Management Science, v 43, n 1, p 88-111

Meyer, MH & Utterback J 1995
‘Product development cycle time and commercial success’, IEEE Transactions on Engineering Management, v 42, n 4, p 297-304

Meyer, MH & Dalal, D 2002
‘Managing platform architectures and manufacturing processes for non-assembled products’, The Journal of Product Innovation Management, v 19, p 277-293
Meyer, MH & Lehnerd, A 1997
The Power of Product Platforms, The Free Press, New York
Miller, TD 2001
‘Modular Engineering – An approach to structuring business with coherent modular architectures of artifacts, activities and knowledge’, PhD Thesis, Technical University of Denmark, Lyngby
Morgan, JM & Liker, JK 2006
The Toyota Product Development System, Productivity Press, New York
Mortensen, NH 1999
‘Design modelling in a Designer’s Workbench – Contribution to a Design Language’, PhD Thesis, Technical University of Denmark, Lyngby
Mortensen, NH, Harlou, U & Andreasen, MM 2005
‘Identification of platform levels in product development’, 15th International Conference on Engineering Design ICED05, Melbourne
Muffatto, M 1998
‘Reorganising for product development: Evidence from Japanese automobile firms’, International Journal of Production Economics, v 56-57, p 483-493
Muffatto, M 1999
‘Introducing a Platform Strategy in Product Development’, International Journal of Production Economics, v 60-61, p 145-153
Muffatto, M & Roveda, M 2002
‘Product architecture and platforms: a conceptual framework’, International Journal of Technology Management, v 24, p 1-16
Nanda, J, Simpson, Kumara & Shooter 2006
‘A Methodology for Product Family Ontology Development Using Formal Concept Analysis and Web Ontology Language’, Journal of Computing and Information Science in Engineering, v 6, p103-113
Nilsson, C 2006a
‘Design reuse experiences from Norwegian manufacturing companies’, 1st Nordic Conference on Product Lifecycle Management - NordPLM’06, Göteborg
Nilsson, C 2006b
‘On the benefit, cost and uncertainty of preparing components for reuse’, 9th International Design Conference Design2006, Dubrovnik
Nilsson, C 2006c
‘Towards uncertainty-aware management of design reuse’, Proceedings of the NordDesign 2006 conference, Reykjavik
Olesen, J 1992
‘Concurrent Development in Manufacturing – based on dispositional mechanisms’, PhD Thesis, Technical University of Denmark, Lyngby
Otto, K & Wood, K 1998
‘Product Evolution: A Reverse Engineering and Redesign Methodology’, Research in Engineering Design, v 10, p 226–243
Pahl, G & Beitz, W 1996
Engineering Design – A systematic approach, Second Edition, Springer-Verlag, Berlin
Pedersen, K, Emblemsvåg, J Allen, J & Mistree, F 2000

‘The validation square – Validating design methods’, ASME Design Theory and Methodology Conference

Prencipe, A & Tell, F 2001
‘Inter-project learning: Processes and outcomes of knowledge codification in project-based firms’, Research Policy, v 30, p 1373-1394

Ramdas, K 2003
‘Managing product variety: An integrative review and research directions’, Production and Operations Management, v 12, n 1, p 79-101

Ramdas, K, Fisher, M and Ulrich, K 2003
‘Managing variety for assembled products: Modeling component systems sharing’, Manufacturing & Service Operations Management, v 5, n 2, p 142-156

Repenning, N 2001
‘Understanding fire fighting in new product development’, The Journal of Product Innovation Management, v 18, p 285-300

Riitahuhta, A 2001
‘Views and experiences of Configuration Management’, in Riitahuhta, Andreasen (eds), Design for Configuration – A debate based on the 5th WDK Workshop on product structuring, Springer, Berlin

Robertson, D & Ulrich, K 1998
‘Planning for product platforms’, Sloan Management Review, v 39, n 4, p 19-31

Sanchez, R & Mahoney, J 1996
‘Modularity, flexibility and knowledge management in product and organisation design’, Strategic Management Journal, v 17, p 63-76

Sanderson, S & Uzumeri, M 1995
‘Managing product families: The case of the Sony Walkman’, Research Policy, v 24, p 761-782

Sanderson, S & Uzumeri, M 1997
Managing product families, Irwin Professional Publishing, Chicago

Schmid, K 2003
‘Integrated cost- and investment models for product family development’, publication by the Fraunhover Institut Experimentelles Software Engineering, downloaded Oct 1st 2004 from http://publica.fraunhofer.de

Schuh, G & Tanner, H 1998
‘Mastering variant variety using the Variant Mode and Effects Analysis’, ASME Design Engineering Technical Conferences DETC98, Atlanta

Seepersad, CC, Pedersen, K, Emblemsvag, J, Bailey, RR, Allen, JK & Mistree, F 2005
‘The Validation Square: How Does One Verify and Validate a Design Method?’, working paper, to be published in K Lewis, W Chen, L Schmidt, (eds.) ‘Decision Making in Engineering Design’, ASME Press, NY, downloaded Sept 30th 2006 from http://www.me.utexas.edu/~ccseepersad/

Sen, A 1997
‘The role of opportunism in the software design reuse process’, IEEE Transactions on Software Engineering, v 23, n 7, p 418-436

Sim, SK & Duffy, AHB 2004
‘Evolving a model of learning in design’, Research in Engineering Design, v 15, p 40–61

Simpson, T 2003
‘Product platform design and optimization: status and promise’, Proceedings of ASME DETC’03, Chicago

Simpson, T, Maier, J & Mistree, F 2001

‘Product platform design: method and application’, Research in engineering design, v 13, p 2-22

Smith, JS, Duffy & AHB 2001
‘Product structuring for design re-use’, in Riitahuhta, Andreasen (eds.), Design for Configuration – A debate based on the 5th WDK Workshop on product structuring, Springer, Berlin

Sobek II, D, Ward, A & Liker, J 1999
‘Toyota’s principles of set-based concurrent engineering’, Sloan Management Review, v 40, n 2, p 67-83

Sosa, M, Eppinger, S & Rowles, C 2003
‘Identifying modular and integrative systems and their impact on design team interactions’, ASME Journal of Mechanical Design, v 125, p 240-252

Starr, M 1990
‘The role of project management in a fast response organization’, Journal of Engineering and Technology Management, v 7, n 2, p 89-110

Suh, N 1998
‘Axiomatic Design Theory for Systems’, Research in Engineering Design, v 10, p 189-209

Sundgren, N 1999
‘Introducing interface management in new product family development’, Journal of Product Innovation Management, v 16, p 40-51

Thornton, A, Donnelly, S & Ertan, B 2000
‘More than just robust design: Why product development organisations still contend with variation and its impact on quality’, Research in Engineering Design, v 12, p 127-143

Tornberg, K, Jämsen, M & Paranko, J 2002
‘Activity-based costing and process modeling for cost-conscious product design: A case studyin a manufacturing company’, International Journal of Production Economics, v 79, p 75-82

Ullman, D 2003
The mechanical design process, Third edition, McGraw-Hill, New York

Ulrich, K 1995
‘The role of product architecture in the manufacturing firm’, Research Policy, v 24, p 419-440

Ulrich, K & Eppinger, S 2003
Product Design and Development, Third edition, McGraw-Hill, New York

Ward, A, Liker, J, Crostiano, J & Sobek II, D 1995
‘The second Toyota paradox: How delaying decisions can make better cars faster’, Sloan Management Review, v 36, n 3, p 43-61

Weber, C, Werner, H, & DEUBEL, T 2003
‘A different view on Product Data Management/Product Life-Cycle Management and its future potentials’, Journal of Engineering Design, v 14, n 4, p 447–464

Yang, B, Burns, N & Backhouse, C 2004
‘Management of uncertainty through postponement’, International Journal of Production Research, v 42, n 6, p 1049-164