INTERNATIONAL DESIGN CONFERENCE - DESIGN 2006
Dubrovnik - Croatia, May 15 - 18, 2006.

DESIGN 2006

ON THE BENEFIT, COST AND UNCERTAINTY OF PREPARING COMPONENTS FOR REUSE

C. Nilsson

Keywords: design reuse, multi-product development, modularisation

1. Introduction

1.1 The need for better understanding of design reuse

One challenge for many companies is to offer a competitive product variety to the market while limiting the costs driven by such variety over successive product generations. This means both to become better (faster, cheaper, more accurate) at launching new products and to reduce the system costs due to *internal variety*. Roughly, internal variety stems from the number and diversity of items and operations (e.g. components and process steps) needed by the company to produce its product portfolio. Internal variety is the main contributor to the so called *complexity* costs [Blackenfelt 2001]. To meet this challenge, companies must exploit synergies between products and product development projects. One way of exploiting synergies is to reuse designs between products. Often the main benefit from design reuse is sought in the *avoidance* of the cost of developing new solutions that would otherwise add to the internal variety costs.

Of course, there is also a cost to reuse solutions, which may in cases be larger than the benefit. Unfortunately, the effects of design reuse often 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.

How can we model "ideal reuse behaviour" to serve as the ultimate goal for reuse strategies? To define *ideal reuse* for this paper, we eliminate the factor of uncertainty by valuing the actual series of products *in retrospect* supposing all the facts (costs) are known. Then are several possible simplifications:

1. One is to suppose that the same technological knowledge is available to the company during the development of all the products in the series, and there is no cost for transferring solutions from one product to another. This is roughly the situation when a product family is developed in one project. Then we consider which solutions that would be optimal with respect to the lifecycle performance/costs of all the products, and determine which solutions that would have been appropriate to share between them.
2. Another is to assume one knows the transaction cost of reuse and 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). Then the ideal reuse could be modelled as the reuse that both minimises the total lifecycle costs of the products *and* the development costs and lead time.

A design reuse strategy has the aim of improving the reuse behaviour of companies over successive products. This does not necessarily imply *more* reuse, but *better* reuse, because design reuse is not an end by itself but a means for companies to offer an attractive variety to the market while keeping costs down. Design reuse is a product-based strategy, aiming to exploit as much value as possible from fewer solutions rather than aiming to improve the capacity to generate and handle more design solutions (process-based strategy) [Fisher et al 1999]. Three questions are central in a reuse strategy:

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 value chain capabilities are known. Fisher argues 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 last of the reuse strategy 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.

Product platform strategies aim to exploit commonality between members of a product family while providing an attractive variety to the market. This is achieved by choosing a product family architecture that determines what should be kept common between variants and what attributes should differentiate the variants, and that optimises the performance/cost ratio for the development and lifecycle of the product family as a whole [Meyer and Lehnerd 1997]. In order for this 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 its value is to be exploited through the creation of derivatives, until the platform itself is upgraded. 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.

While "platform-thinking" is a sound mindset applicable to many settings, most *platform strategies* as discussed in literature (e.g. [Meyer and Lehnerd, 1997], [Gonzalez-Zugasti 2001], [Krishnan and Gupta 2001]) are top-down in nature and require company-wide realignments. These 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 multi-product performance. Therefore, it is here argued that there is a need for "bottom-up" models that improve the understanding of how low-level design decisions (here with focus on reuse) affect the long term multi-product development performance.

This paper intends to explore what the basic elements and mechanisms of design reuse are, including the main expectable benefits and costs. Then, the issue of preparing solutions for future reuse will be targeted with the intent of formulating the problem area as a basis for further research: Which are the elements and main issues of a proactive reuse approach? The paper is mainly based on literature on multi-product development.

1.2 Theoretical background

Here, a *design solution* is seen as an actively chosen solution to a design problem (also called design parameter). The design of a product often consists of innumerable design solutions. A design solution 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 granularity: 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 lenght 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.

Design solutions must be more than an idea to be useful. In a company, design solutions are normally "embodied" in a number of assets:

- requirements (the solved design problem)
- communication/documentation of the solution itself, specifications
- implementations (e.g. components, prototypes), virtual models, test results
- adaptations/dispositions: to cope with a design solution, different stages of the value chain often have to adapt in one or another way, e.g. tuning of production machinery
- knowledge, explicit and tacit, about the reasoning and experiences behind the solution

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 the value chain. 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 the value chain.

Here, the focus is on *reuse* of design solutions between successive products, as opposed to *sharing* of solutions across concurrent products. The most relevant *reuse-related activities* in this context are:

- reusing without modifications (e.g. identical components)
- reusing with modifications (principles/parameterised design)
- preparing solutions for reuse

These reuse activities can be seen as complemented by the so called modular operators: splitting; substituting; augmenting; excluding; inverting; and porting [Baldwin and Clark, 2000]. While "reusing without modifications" is analogous to a null or *no-change* modular operator, "reusing with modifications" is analogous to *porting* and "preparing solutions for reuse" to *inverting*.

For practical reuse of a part of a product, its interfaces must be explicit and its behaviour predictable. With these criteria, a part that has hidden dependencies to other parts is not reusable. A *reusable unit* is a part or a minimal group of interdependent parts with well-defined explicit interfaces and functions. A reusable unit contains the notion of internal interdependency and external independency, which coincides with what some authors call a *module* [e.g. Baldwin and Clark 2000]. Therefore, to make interdependent parts reusable, either these 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. 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.

The term *component* is in this paper used to denote a reusable unit, that is, a physical part or a group of interdependent parts with well-defined external interfaces and functions.

1.3 Benefits and costs of design reuse

The main expected benefits of design reuse are related to

- *Product development*. Design reuse can save product development resources, because of the avoided design and test costs of introducing a new design.
- *Production*. Increasing the commonality between products can streamline the supply chain, yielding increased economies of scale and reducing "*the required production investment associated with a new product*" [Fisher et al 1999, p.298].
- *Internal variety*. If fewer newer solutions are introduced, the assortment should be easier to overview and manage, thus reducing the complexity costs of the company.

The two main types of design-reuse driven costs are:

- *Transfer cost*. The cost of finding, analysing and incorporating a previous solution. This cost is mostly related to the capture and transfer of information. The costs of *preparing* components for reuse are mainly related to the invested development efforts.

- *Overdesign*. Component reuse can in some cases increase production costs or reduce technical performance because of overdesign, due to the design compromise that may be necessary to enable a component to be used both in a high-end and a low-end product [Krishnan and Gupta 2001]. 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]

Of course, there are other costs indirectly related to design reuse. For example, business-strategic disadvantages can stem from inadequate reuse causing technological stagnation. This is for instance addressed in literature about innovation strategies.

2. Preparing components for reuse

A key for companies to reduce their product development-related waste is to be *proactive* regarding design reuse, i.e. investing to improve the future possibility to reuse solutions. Of course, preparation for reuse does not have to be all or nothing, since discrete increments may be possible ("steps" in an effort/benefit curve). For example, the effort of thorough testing of one component may bring extra value to a future project if it reuses the component. But there is a minimum level of reusability that has to be reached for the solution to be convenient to reuse at all. If the preparation does not reach that level, it could turn out to be wasted. The focus here is on reuse of discrete *components* but it is believed that the discussion also is applicable to more abstract designs, like parameterised designs.

2.1 A reuse strategy

A reuse strategy must select which components the company should reuse over time based on business-strategic factors. Due to the investments related to design reuse, reuse management must deal with the allocation of resources over time, and is in this sense a form of portfolio management.

Reuse strategies should take advantage of the *relative* maturity of the solutions, so that development efforts are spent where they are most needed. More precisely, it is the expected rate of evolution of the solutions compared to the rate of new product launches that matter. Some solutions may become obsolete by the time of there is an opportunity of reuse them. For example, it may show unwise to expect an experimental component to be reused unchanged if its improvement potential increases rapidly thus making it more convenient to redesign it by the next project. Other solutions may be sufficiently stable to encourage "mass reuse". Meyer and Lehnerd highlight the importance of being selective when choosing what to reuse:

> *Standardization for all components across a product portfolio can lead to rigidity in underlying platforms and inflexibility when it comes to advancing product designs into the future. [...] An effective approach towards standardization is one that is highly selective, carefully choosing elements that should be standardized.* [Meyer and Lehnerd 1997, p.121]

Fisher proposes the categorisation of components according to their influence on "product quality" (customer perception of product performance), further arguing that for components of weak influence, the cost issues should determine their reuse policy [Fisher et al 1999]. Some strategically important solutions at the technological forefront should perhaps never be reused unmodified because it is vital for the company that they are under continuous evolution.

2.2 Assessing the reuse potential of components

In order to select and prepare components for reuse, it is necessary to forecast future product requirements. This can be done by analysing a hypothetical product family based on the company's road map, and try to identify an optimal set of components to reuse (ideal reuse) [Fisher et al, 1999].

The following factors affect the likeliness a given solution will be beneficial to reuse in the future:

- Probability that the design problem will reappear in future (product specific need?)
- Quality of solution: how well does the solution solve the design problem? How high will the improvement potential be in the future (e.g. with increased technological knowledge)?

- What will be the cost to reuse the solution? For example, will it consume resources to find and understand the solution? What will be the performance penalty if the design turns out to be suboptimal (e.g. over-dimensioned)?
- What will be the benefits of reusing the solution, i.e. avoiding the cost of introducing a new solution? (both in terms of product development and solution lifecycle costs)
- Possibility/need of delegating/postponing reuse decisions (management/risk dimension)

Of course, here the potential to *improve* reusability should be considered. Is it possible to make the solution significantly more reusable through small efforts and/or technical compromises?

The complexity of a component's interfaces is an important factor determining its reusability. Obviously, for correct reuse a component's interfaces have to be explicit, that is, documented or self-evident. Unnecessary limitations on flexibility should be avoided by clever design. Some interfaces may be more "normative" than others, i.e. the range of settings in which the component may be reused has been limited beforehand. Examples of this can be observed in some product platform implementations, where the derivatives are predicted to a degree during the design of the platform [Gonzalez-Zugasti 2001]. Other components may allow much more reuse flexibility because their interfaces require few matching design parameters. These can be reused in many unpredicted settings. Of course, such elegant solutions are not always possible to find.

Another requirement for a component to be reusable is that the reusers know how to reuse it. Note that the reusers do not need to know how to design it, because "*seen in relation to product development and product life, we have to make a distinction between the knowledge, which created the solutions and the knowledge, which is necessary for the utilisation*" [Andreasen et al 2001, p.29]. Obviously, companies normally do not possess the know-how to redesign all the solutions they use. Most often this is no problem as these solutions can be reused without modifications. However, in other cases designers must redesign the solutions, hence needing a deeper technological understanding. Therefore, it is necessary to view the reusers' needs for each reuse situation. Typical reuse information is a detailed description of the component's interfaces and functions, performance graphs, test results, along with restrictions and tips. Finally, references to deeper sources of knowledge should be available. (In industry, it sometimes seems like the most important reuse information is the phone number of the engineer that designed the solution.)

2.3 Preparing a component for reuse

Because reusable components have to be designed to fit in a range of future products and be correctly reused by the future designers, the following additional design activities are needed:

- Forecast future product requirements to specify component reuse requirements
- Design well-defined interfaces and robust performance, following modularisation principles, e.g. by isolating the stable from the unstable and reducing complexity
- By preparing components for reuse, designers set "design rules" [Baldwin and Clark 2000] or "game rules" [Andreasen et al 2001] that the future re-users must follow to reuse the components in harmony with the product architecture. For example, product platform strategies often imply that those designing the product platform impose a set of design rules for the development of variants. Therefore, designing for reuse may demand a significantly higher architectural knowledge (understanding of the solution's role in the life cycle of the product and business context) than designing a non-reusable solution.
- Invest in higher quality, considering the possibility of reuse in several products and the following higher volumes. Fisher has observed that this 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, p299]

In addition, the preparation should include proper documentation for reuse. It is vital for the benefits of reuse to be collected that the information needed by the re-users is retrievable in an efficient way. Software tools like PDM/PLM systems 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. One challenge is how future reuse planners should be informed when there exists a reusable solution they could 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. 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.

2.4 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. Uncertainty handling of course becomes more delicate the higher the investment involved. 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 component reuse are related to:

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

Uncertainty can in many cases be considerably reduced through the analysis of existing information like historical data. The manner in which a company acts upon the remaining uncertainty is influenced by how well the company perceives it can predict the future, how optimistic or pessimistic it is, and which attitude the company has upon ambiguity: "aggressive" (investing heavily and taking risks in pursue of possible benefits), "defensive" (preferring not to invest for ambiguous outcomes) or "explorative" (e.g. investing in pilot projects).

A tough management decision ridden with uncertainty is whether to invest in preparing solutions for future reuse. How much is it worth (today) adding the possibility to reuse certain solutions in a future project? A *real options* method, based on option pricing theory, has been proposed to measure the expected value of design options:

> *The basic concept is the following: 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]

2.5 Organisational issues

In this final section I would like to comment on the organisational issues that might affect the reuse behaviour of companies. First, there 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].

Second, the question of who takes the decisions and who has the "burden of proof" about long-term considerations as component reuse may be problematic. This may show in heavyweight project

organisations, where coordination with other functions in the company may be more difficult than in functional organisations [Fisher et al 1999].
Third, one may ask when component reuse should be promoted incrementally, bottom-up, and when it requires a company-wide commitment, for example because they depend on organisational changes and managerial commitment to be beneficial. An example of a seemingly appropriate bottom-up decided reuse could be an elimination of an "obvious" component duplicate. An example where a top-down decision flow seems appropriate could be case of the product platform program at Black&Decker, analysed by Meyer and Lehnerd [1997] that conclude that the success of the program to a great extent was due to the holistic approach and commitment of the management.

3. Conclusions

The issue of design reuse is complex as it implies effects that are difficult to quantify and show in the long term. If future product needs are predictable, and resources are available to invest, then a planned reuse strategy can be applied where a product family architecture is laid out deciding what is to be reused over a planned family of products, following a product platform approach. In this case the choice of what to reuse is mostly a demanding but conceptually straightforward optimisation exercise.
However, for companies that cannot predict future products, the reuse approach must be managed continually and incrementally product generation after product generation. Such reuse management must consider what the upcoming product should reuse from the past, and what the ongoing product development project has to prepare for future reuse. One step towards a general reuse strategy is to formulate the envisioned ideal reuse behaviour for the company. Reuse management should also identify when the reuse of given components has such a strategic impact that it should be required explicitly from a certain project, as opposed to delegating the decision to the project.
The following issues have been identified as central in relation to proactive component reuse:

- *Preparing a component for reuse* implies designing the component to meet a set of expected reusability requirements, for example dimensioning the component to function with predicted future products. Modularisation methods are very useful to this end. Furthermore, the preparation implies visualising the "reuser" as an additional customer to design for. For example, the reuser may require documentation on how to reuse the component, and/or clarity of design to improve the chances the component gets correctly reused.
- *The benefits* of such preparation are (of course) increased likelihood of the components being reused, which in turn can yield various benefits to the future project(s), like decreased technical risks, reduced designing and testing needs, increased production volumes, etc. The short term benefit to the project where the preparation is done can be improved design quality resulting from the extra designing and testing effort.
- *The costs* of the preparation are on the one hand the development resources diverted from other project activities, and the possible overdesign of the components that can result from the adaptation for future products.
- *The uncertainties* regard primarily the forecasting of the reusability requirements, and how flexible for reuse the component will be after preparation. Option pricing thinking can support the valuation of the expected benefits in order to prioritise the allocation of designing efforts.
- *Selecting which components to prepare* is a delicate exercise where issues like the maturity of the solutions and the technological knowledge behind them must be considered along with company strategy and customer perception of product quality.

References

Andreasen, M.M., McAloone, T, Mortensen, N.H, "P Insight report on Multi-product development – platforms and modularisation", Dept of Mechanical Engineering, Technical University of Denmark, Lyngby 2001*

Baldwin, Clark, "Design Rules: The Power of Modularity", The MIT Press, Massachussets, 2000

Blackenfelt, "Managing complexity by product modularisation: Balancing the aspects of technology and business during the design process", Doctoral Thesis, Royal Institute of Technology, Stockholm, 2001

Fisher, M, Ramdas, K, Ulrich, K, "Component Sharing in the Management of Product Variety", Management Science, Vol. 45, No. 3, 1999, pp 297-315

Ericsson, Erixon, "Controlling Design Variants", Society of Manufacturing Engineers, Michigan, 1999
Gonzalez-Zugasti, J, Otto, K, Baker, J, "Assessing value in platformed product family design", Research in Engineering Design Vol. 13, No. 1, 2001, pp. 30-41
Krishnan, V, Gupta, S, "Appropriateness and impact of platform-based product development", Management Science Vol. 47 No. 1, 2001, pp. 52-68
Meyer, M.H., Lehnerd, A.P., "The Power of Product Platforms", The Free Press, New York, 1997

Cristian Nilsson, M.Sc. Mechanical Engineering
PhD student
Norwegian University of Science and Technology, Department for Engineering Design and Materials
Richard Birkelandsvei 2B, 7491 Trondheim, Norway
Tel.: +47 73593816
Email: cristian.nilsson@ntnu.no