Chapter 13

A ROADMAP FOR PRODUCT ARCHITECTURE COSTING

Sebastian K. Fixson
Department of Industrial & Operations Engineering, University of Michigan, Ann Arbor, MI 48109

1. INTRODUCTION

In recent years many markets have exhibited increasing demand heterogeneity; they are fragmenting into more and smaller market niches. This development threatens the large-scale assumption of many mass production processes. As a result, firms face the dilemma of how to provide a wide variety of goods for prices that can compete with mass produced products. To respond to these challenges, many firms have begun searching for ways to combine the efficiency of mass production with the variety of customer-oriented product offerings. A major focus of these efforts has been the fundamental structure of the product: the product architecture. Examples for this development are Sony's personal music players (Walkman) that use common drives across different models (Sanderson and Uzumeri, 1995), different power tools that use similar motors (Meyer and Lehnerd, 1997), PDAs (personal digital assistant) that can be turned into an MP3 player, a camera, or a telephone with different attachments (Biersdorfer, 2001), and automobiles with common components across models (Carney, 2004).

Researchers of disciplines ranging from engineering to management have focused their attention on these phenomena, and have developed tools to guide the difficult process of providing variety to the customer while maintaining near-mass production efficiency, i.e., to 'mass customize' (Pine, 1993a). The approaches vary in their perspective and level of analysis. Some focus more on ways to increase external product variety while

T. W. Simpson et al. (eds.), *Product Platform and Product Family Design*

maintaining low costs, while others target their efforts on internal variety reduction without losing the variety appeal for the customer. The underlying idea of most of these approaches is to increase commonality across multiple products. The level in the product hierarchy at which commonality is pursued varies: it can be focused on common components (Eynan and Rosenblatt, 1996; Fisher, et al., 1999), on modules (Chakravarty and Balakrishnan, 2001; Dahmus, et al., 2001; Sudjianto and Otto, 2001), on product platforms and product families (Gonzalez-Zugasti, et al., 2000; Jiao and Tseng, 2000; Simpson, et al., 2001) or on production processes (Wilhelm, 1997; Siddique, et al., 1998), although the lines between these levels are sometimes blurred.

From an overall strategic perspective a firm needs to balance all benefits it can achieve by increasing commonality across products with all the costs this approach creates. For example, it needs to weigh the revenue decreasing effects through cannibalization that product commonality can cause against the cost savings that commonality can achieve (Robertson and Ulrich, 1998). Ideally, this multi-objective problem requires the balancing of cost, revenue, and performance effects when selecting a product architecture from a set of candidates. Although cost is only one of these variables, there are at least two major reasons that make it worthwhile to explore this problem with a focus on the cost portion alone. The first reason is that in many cases cost is a major, if not the most important, decision variable. More specifically, most products contain two types of components: those with a strong influence on product quality and those with only a weak influence on product quality (Fisher, et al., 1999). For components of the latter type cost becomes the only decision variable, provided that the components' performance level is sufficient (Thonemann and Brandeau, 2000). The second reason for building a roadmap focusing on cost is that it can—once established—serve as a building block for the development of more sophisticated design support tools such as product architecture design guidelines or optimization models. These tools often build on existing cost estimation models which in turn incorporate known or assumed relationships between product architecture and costs as well as cost allocation rules, and to interpret the results of the models requires a thorough understanding of how the problem has been framed. In other words, what are the multiple and complex relationships between various product architecture characteristics and various costs along the product life cycle? The existing research is somewhat inconclusive. For commonality decisions, one aspect of product architecture, effects on individual costs have been demonstrated (Park and Simpson, 2003), whereas for modularity, another aspect of product architecture, no general relationship with cost has been found (Zhang and Gershenson, 2003). In other words, the complex relationship between product architecture and costs is still insufficiently understood (Simpson, 2004). Similarly, what is the impact of the applied allocation rules on the cost models, and consequently, their results? Finally, to develop the deep understanding of the relationships between product architecture and costs in turn requires a good understanding of the input data, i.e., how is the product architecture described and what types of costs are considered? Figure 13-1 illustrates this chain of requirements for building design support tools with respect to costs. The remainder of the chapter develops a roadmap that helps covering all requirements from input data to the cost estimation models.

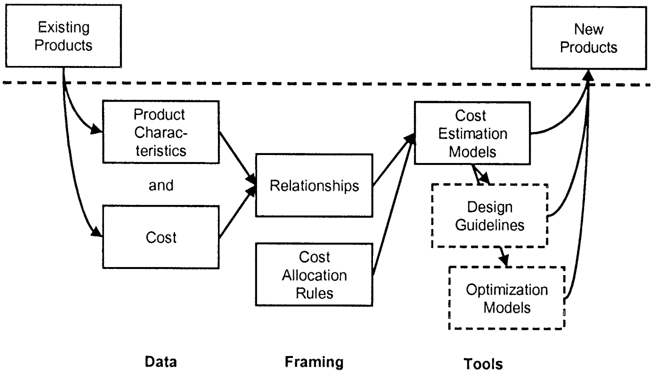


Figure 13-1. Requirements chain for developing product design support tools for cost.

2. DEVELOPING A ROADMAP FOR PRODUCT ARCHITECTURE COSTING

The roadmap comprises four steps (see Figure 13-2). The first step is an assessment of the differences in product architecture between potential candidates. This step is crucial because in order to make the analysis of cost consequences of different product architectures possible requires the ability to distinguish different product architectures in the first place. The product architecture costing roadmap builds on a multidimensional product architecture description methodology. In the second step of the roadmap the relevant life cycle phase, or phases, with respect to costs have to be identified. The question of relevance hinges on a variety of factors such as product lifetime, production volume, total value, and cost ownership. The third step requires determining the cost allocation rules to be used for the costing procedure. The choice of certain accounting decisions can have a profound effect on how the product architecture-cost relationship is modeled. In its fourth step, the roadmap calls for the selection of suitable cost models. Existing models differ in their requirements for data accuracy and sample size, as well as their ability to predict cost differentials of product architectures differences. Each step of the proposed roadmap is discussed in more detail in the following sections.

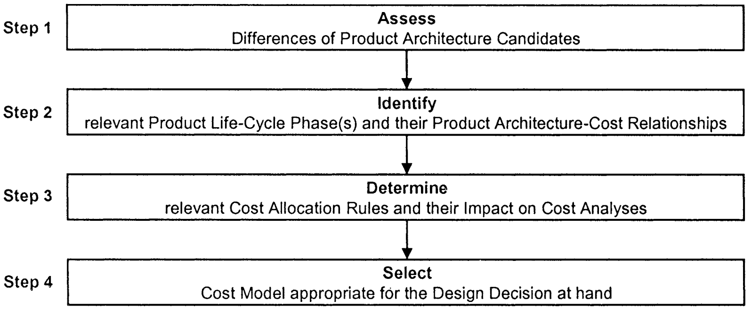


Figure 13-2. A roadmap for product architecture costing.

2.1 Step 1: Assess differences of product architecture candidates

2.1.1 The special role of product architecture as a design variable

Product designers make numerous decisions throughout the design process. Each of these decisions has consequences for some costs along the product life cycle. Two characteristics label the links between these decisions and their cost consequences. The first characteristic describes how difficult it is to construct the link; the second how valuable it is to know it (see Figure 13-3).

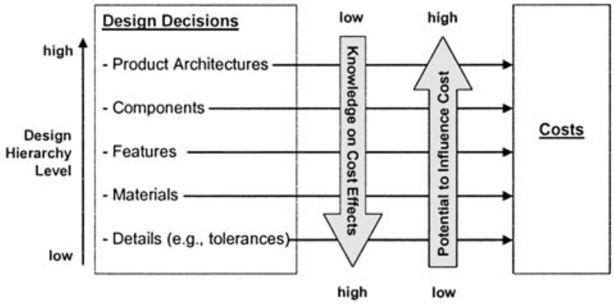


Figure 13-3. Product architecture decisions in the design hierarchy.

The level of difficulty to establish a link between design decisions and their cost effects depends on the hierarchy level at which the decisions are made. On a very detailed level, it is fairly straightforward to construct a link between the design decision and its cost implication for two reasons. First, on the detailed level it is often clear on what costs to focus on, and second, for well known links historical data often exist. For example, there is ample data on how a more stringent surface smoothness requirement affects the manufacturing cost to create that surface. Design textbooks typically provide cost tables or functions to guide these type of design decisions (Michaels and Woods, 1989; Pahl and Beitz, 1996). On the next higher level of abstraction, design decisions affect the choice of materials, production processes, or part features. Materials have been used as a cost determining decision variable for a long time since in many mass production environments material costs represent a significant fraction of total production costs (Ostwald and McLaren, 2004). For this reason, rules-of-thumb have been developed to allow approximate but quick cost estimates. For example, to assess the cost impact of selecting manufacturing processes, Esawi and Ashby (2003) have developed a simple model that requires the input of only a few parameters. The primary aim of that method, however, is the relative ranking of multiple processes with respect to cost, not to predict exact costs. Product features have also been used as decision variables for which cost models have been developed. Often the models combine cost estimations on the feature level with cost estimations on component and assembly levels (Weustink, et al., 2000) and the product family level (Park and Simpson, 2003). Yet another level up in abstraction is populated by design guidelines such as Design for Manufacturing (DFM) or Design for Assembly (DFA). They represent codified knowledge of links between design decisions and production costs. However, they are not cost prediction tools but present the knowledge in a condensed form such that they direct the designer's attention to cost creating design issues, and lead him towards (relatively) lower cost solutions (Boothroyd, et al., 2002). Finally, on the level of product architecture there are numerous examples for relationships between *individual* aspects of the product architecture and *individual* costs, but no approach exists that provides a generic yet comprehensive description of this multidimensional relationship.

The second characteristic that describes the link between design decisions and cost elements along the product life cycle is the leverage to influence the costs if the link is actually known. It is generally assumed that earlier design decisions have greater potential to influence costs than later design decisions.[18] This creates a dilemma for the designer: it is the early phase of design decisions where the potential to influence total life cycle cost is the greatest, and yet early in the design process is where the fewest data for detailed cost estimations exist. How, then, can this link be constructed? This roadmap builds on a methodology that can distinguish between different product architectures along multiple dimensions on a relatively abstract level.

2.1.2 A multidimensional method to assess product architecture differences

The product architecture is the fundamental structure and layout of a product and is defined during concept development (Ulrich and Eppinger, 2004). Building on Ulrich's description of product architectures (Ulrich, 1995), a multi-dimensional product architecture description method has been developed (Fixson, 2005). The method relaxes three fundamental assumptions of earlier work. First, it allows for independent assessments of the two main product architecture dimensions: *function-component allocation* and *interfaces*. Second, it acknowledges that these two dimensions are themselves multidimensional constructs. Third, it assesses the product architecture for each function separately—in contrast to most product architecture descriptions in the literature that essentially provide *average* assessments of a product's architecture.

The first of the two dimensions, *function-component allocation (FCA),* is concerned with the extent to which a product's functions are isolated on physical components. It measures for each function (on the selected architecture level) the degree of function-component allocation. More specifically, each function is assigned two indices that determine its position relative to the extremes of 1-to-1 and many-to-many relationships between functions and components. A 1-to-1 measurement indicates a situation in which the function under consideration is provided exclusively by one component, and this component provides exclusively this function only. This style of FCA is called modular-like. In contrast, a few-to-many measurement indicates a situation in which a function is provided by many components (an integral-fragmented style). A many-to-few measurement

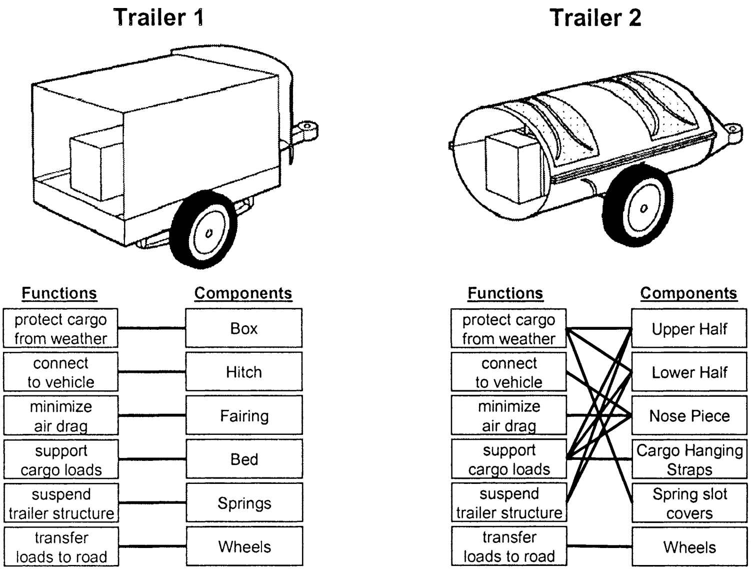
---

[18] Various authors present the idea that somewhere between 60% and 90% of the total life cycle cost are committed during product design. Interestingly, although these numbers are used by a variety of authors from diverse fields ranging from accounting to engineering to management, e.g., (Smith and Reinertsen, 1991:100; Anderson and Sedatole, 1998:231; Blanchard and Fabrycky, 1998:561; Clancy, 1998:25; Knight 1998:21; Sands, et al., 1998:118; Buede 2000:7; Weustink, et al., 2000:1; Bhimani and Muelder, 2001:28), nowhere is real data presented as evidence. One exception exists that models costs in more detail, however, it also does not specify a particular fraction of the total life cycle cost that is committed during design, but rather assesses the cost influence potential of the design phase versus the one of the production phase (Ulrich and Pearson, 1998).

denotes an integral-consolidated style where one component provides multiple functions. Finally, a many-to-many measurement represents an integral-complex FCA style. It is important to take the FCA measure for each product function individually because the reuse of a component across a product family depends to a large degree on the role a component can play in different products. The second dimension of the product architecture description method, *interfaces,* is itself multidimensional and is concerned with three characteristics of the interfaces that connect the components. The first characteristic, interface intensity, describes in detail the role each interface plays for the product function. Interfaces can be spatial, or they can transmit material, energy, or signals or any combination of the above (Pimmler and Eppinger, 1994). The second characteristic, interface reversibility, describes the effort it requires to disconnect the interface. This effort depends on two factors: the difficulty to physically disconnect the interface, and the interface's position in the overall product architecture. Finally, the third characteristic, interface standardization, depends both on product features as well as the population of alternatives. While some researchers have used different types of interfaces to categorize types of modularity like swapping, sharing, bus, and sectional (Ulrich and Tung, 1991; Pine, 1993a), the method presented here views the extent to which an interface allows different kinds of interchangeability as a matter of perspective. In other words, the level of standardization can be different for any component that is involved in the interface. Standardization is a function of the number of alternatives that exist on either side of the interface.

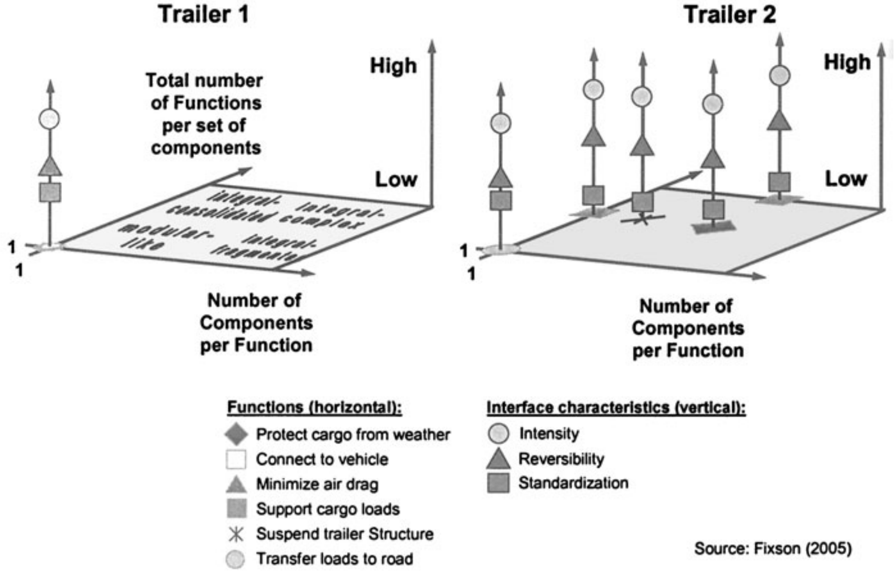
As example, compare the two trailers in Figure 13-4 (top). They both provide the same functionality. However, they exhibit very different product architectures. Figure 13-4 (bottom) shows two different patterns of how each component provides one or more functions. Figure 13-5 illustrates the same information with the help of product architecture maps. Each function is assessed separately and along multiple dimensions. The location on the x-y plane identifies where each function is positioned in between the extreme points of 1-to-1 and many-to-many function-component allocations. To put it differently, the position describes each function's FCA style. The three interface sub-dimensions (intensity, reversibility, and standardization) are independently scaled on the vertical axis. The value of this assessment independence can be seen by comparing individual functions for the two trailers. For example, the function *transfer loads to road* exhibits identical product architecture characteristics for both trailers whereas all other functions show significant differences along the multiple dimensions.

The following sections refer to these dimensions of product architecture when discussing the remaining three steps of the roadmap for product architecture costing: identifying the relevant life cycle phases and costs and their product architecture-cost relationships, determining the relevant allocation rules, and selecting appropriate cost models.



Source: Ulrich (1995) "The role of product architecture in the manufacturing firm"

Figure 13-4. Two trailers with different product architecture.



Source: Fixson (2005)

Figure 13-5. Product architecture maps for the two trailers.

2.2 Step 2: Identify relevant product life cycle phase(s) and their product architecture-cost relationships

Before one can begin to investigate the cost implications of differences in product architecture one has to decide on which costs to incorporate in the analysis. This problem has two components. The first is concerned with the decision of which life cycle phase(s) are relevant with respect to cost for the decision at hand, and the second strives to identify the relevant product architecture-cost *relationships* within the selected life cycle phase or phases. The factors used to identify relevant product life cycle phases are discussed next, followed by a detailed account of known effects that individual product architecture characteristics have on costs for each life cycle phase.

2.2.1 Which life cycle phase matters?

Every product and system, regardless of size, value and lifetime, progresses through different phases during its life: design, development, production, use, and retirement. In each of these phases, different processes and activities are performed with and on the product (see Figure 13-6). Each of these processes and activities creates a cost that occur at different points in time, at different locations, and can be borne by different constituents.

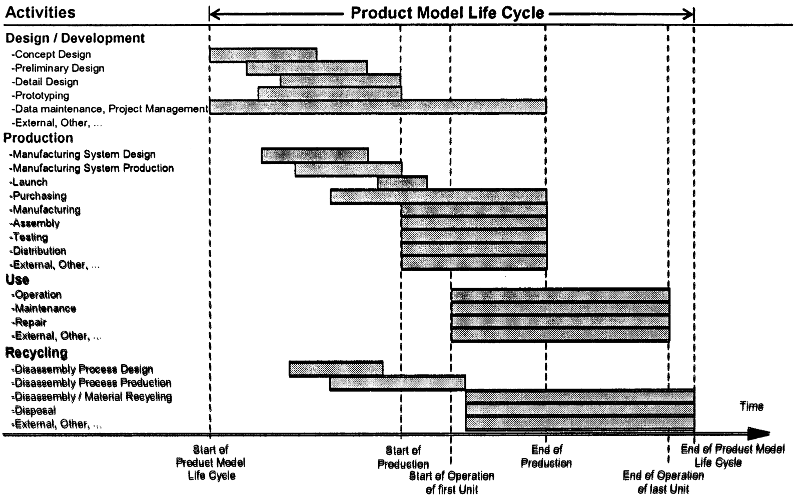


Figure 13-6. Activities throughout the product model life cycle[19].

---

[19] Note that the diagram depicts the product life cycle of all units produced during a model's life. In case of only one unit produced (e.g., expensive or special equipment), the diagram collapses into the individual product's life cycle. In this chapter, the term life cycle refers to the life of a single product.

Given that various costs occur in different phases of a product's life, one of the first decisions of a cost evaluation is to determine those costs that are relevant for the design decision at hand. The relevance of individual costs depends on the life cycle cost profile and the ownership of the costs. A product's life cycle cost profile is determined by both absolute values and relative distributions of the costs over the life cycle, the durations of the individual phases, and the production volume. To separate products according to their absolute values of total lifetime and total life cycle costs, it has been suggested to cluster the universe of different products into three major categories: large-scale, medium-scale and small-scale systems (Asiedu and Gu, 1998). Large-scale systems can have total lifetimes of several decades and total life cycle costs of billions of dollars. Lifetimes of medium-sized products are typically measured in years, with total life cycle costs ranging from thousands to millions of dollars. Small-scale products can have lifetimes as short as a few months and life cycle costs as low as a few hundred dollars (see Figure 13-7).

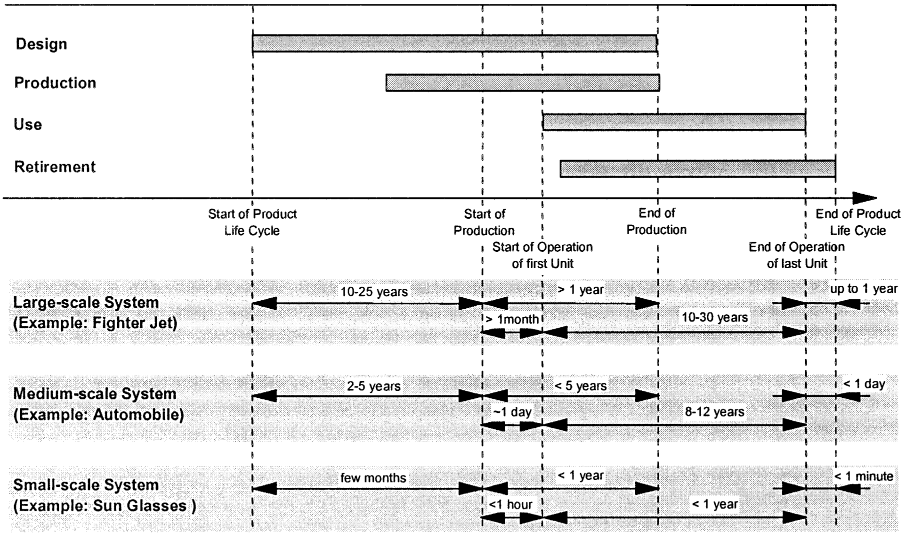


Figure 13-7. Lifetimes of different product categories.

In addition to the absolute values, the relative distribution of time and cost over the different life cycle phases also plays an important role in determining on which costs to focus. These differences in relative distribution can be caused by differences in scale and technical complexity. For example, a small product, say a radio clock, will require very few maintenance and support activities, which translate into low costs during its use, whereas for long living and large scale products as, for instance, a navy ship, these costs can represent almost 2/3 of total lifetime costs (Sands, et al., 1998). Another factor that influences the relative size of the costs of the individual life cycle phases is the production volume per model. A small production volume results in relatively higher development cost per unit compared to the situation in which the total development costs can be spread over a large production volume. The consequence of the differences in total life cycle cost, total life time, life cycle cost distributions, and production volumes are different life cycle cost incurrence curves (see Figure 13-8).

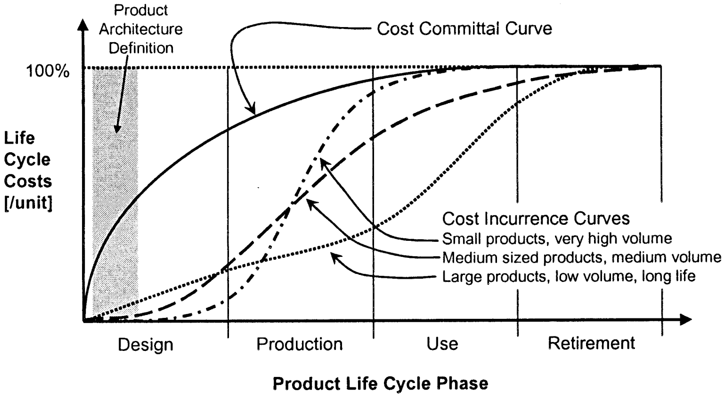


Figure 13-8. Cost committal and incurrence curves.

Finally, the life cycle phase in which certain costs occur does not necessarily determine who bears these costs. For example, warranty policies can transfer costs between producer and user (Blischke and Murthy, 1994), and most of so-called external costs are often borne by the society at large while the product user pays only a fraction of it directly. More generally, depending on a variety of additional factors such as market dynamics, level of competition, or institutional environment, a number of different cost distribution schemes are conceivable, enforced by different contractual agreements. Since most of these factors are not decision variables for the designer, the following discussion of each life cycle phase individually looks at costs independently from the ultimate ownership. Also note that while the primary focus of this chapter is on cost effects triggered by product architecture design decisions, other performance measures—such as time, and to some extent revenue—that are impacted by these decision, are discussed where relevant.

2.2.2 Product architecture effects on costs of the product development phase

The first phase of a product's life encompasses activities such as conceptual and preliminary design, detail design and prototyping, testing, as well as supporting functions such as data maintenance and project management. For engineered products, the costs for these processes represent primarily engineering resources, i.e., personnel. To address the question of how differences in product architecture affect the resource consumption during the design phase some researchers have linked the task structure of the design process to the product architecture (von Hippel, 1990; Eppinger, et al., 1994; Gulati and Eppinger, 1996). Over time, a firm's organizational structure often mirrors the product structure of the products the firm produces (Henderson and Clark, 1990). Thus, the design decision on the number and size of 'chunks' (subsystems, modules, parts, etc.), i.e., the *function-component allocation scheme*, translates into the number and size of teams working to develop the product (Baldwin and Clark, 2000). The number and size of the teams determines their internal complexity as well as their external communication requirements. Both factors in turn determine the teams' efficiency. Either extreme, i.e., one very large team or many, very small teams, appears to be a relatively inefficient organizational form, the former requiring many internal iterations, the latter producing a long sequence of information transfers. Therefore, creating product architectures that balance the design complexity that incurs between the chunks (integration effort) on one hand, with the sum of the design complexity within the chunks on the other, by designing chunks of medium complexity, seems to be a resource efficient approach. This effect has been found empirically for complex software development projects (MacCormack, et al., 2004). For the second product development performance measure next to cost, total development time, a similar effect has been demonstrated: for the development of a turbopump of a rocket engine it has been shown that there is a number of blocks of the product architecture (modules, chunks, etc.) that translates into a medium number of teams that minimizes the duration of the project development project (Ahmadi, et al., 2001). Apparently, both costs and time functions exhibit a minimum if the product is decomposed into a medium number of subunits; and increases when fewer but larger subunits are chosen, and increases when more but smaller subunits are selected.

The relative value of time compared to cost depends on a number of market parameters as well as the ratio between revenue and costs. For example, companies operating in fast pace market environments will especially value a product architecture's potential to reduce the time-to-market. Product architectures that allow conducting much of the design process in clusters in parallel to arrive at the shortest possible total design time are of particular value to them. In a specific case about a Polaroid camera housing, for example, it has been found that the foregone sales in case of a longer development time far outweigh any achievable cost savings in manufacturing (Ulrich and Pearson, 1993). In a case like this, a product architecture that helps to reduce development time is much more valuable than one that focuses on cost savings in the production phase.

Also, strictly speaking, the design phase is only one component of the time-to-market. If 'market' is understood as sale (or start of operation) of the first unit, then production preparations become part of the time-to-market, in particular tool design and manufacturing. Hu and Poli (1997) have compared assemblies made from stampings with injection molded parts regarding their effects on time-to-market. They find that parts consolidation, i.e., the reduction of the number of chunks the product consists of can be disadvantageous with regards to time-to-market when the time to produce the tool for larger, more complex parts extends the total time-to-market.

In addition to the particular product function-component allocation scheme, the characteristics of the *interfaces* between the chunks are likely to affect the efficiency of the design process, and thus its costs. The weaker the interface connections are, i.e., the lower their intensity, the more the different design teams can be working independently on different subsets of the product. This can reduce the number of iterations between the teams, and thus increase overall design process efficiency. In a case study of the development of an automotive climate control system, strong coupling between components has been identified as one reason for development cost increases (Terwiesch and Loch, 1999). Weaker interface dependencies may also improve the second performance indicator, total development time (time-to-market), because it allows the design tasks to proceed in parallel (Baldwin and Clark, 2000). For example, analyzing the product development of integrated circuits it has been found that higher levels of interface independence increase the design flexibility and reduce the risk of having to repeat experiments (Thomke, 1997).

Finally, both characteristics of a product's architecture, i.e., its function-component allocation and its interfaces, affect development costs in a particular way as a consequence of the nature of design work. Design costs are one-time costs in the product model's life, i.e., their relative contribution to the unit costs is highly sensitive to changes in the production volume. If only one product is ever produced, say, a racing boat, then this single unit has to bear all development costs which makes the cost for design and development a substantial portion of the unit's life cycle costs. In contrast, for mass-produced products like vacuum cleaners the design costs are shared by potentially millions of identical products, which makes the design and development costs per unit relatively small. For the assessment of cost implications of architectural decisions this issue is also relevant when product architectures allow sharing of portions (platforms, modules, components) of a product across product families, and, therefore, allow the sharing of their development costs. The savings through the reuse of designs affect both development cost and time (Nobeoka and Cusumano, 1995; Reinertsen, 1997; Siddique, 2001; Siddique and Repphun, 2001). Figure 13-9 shows the mechanisms by which individual (and combinations of) product architecture characteristics express their relationship to performance measures such as product development costs and time.

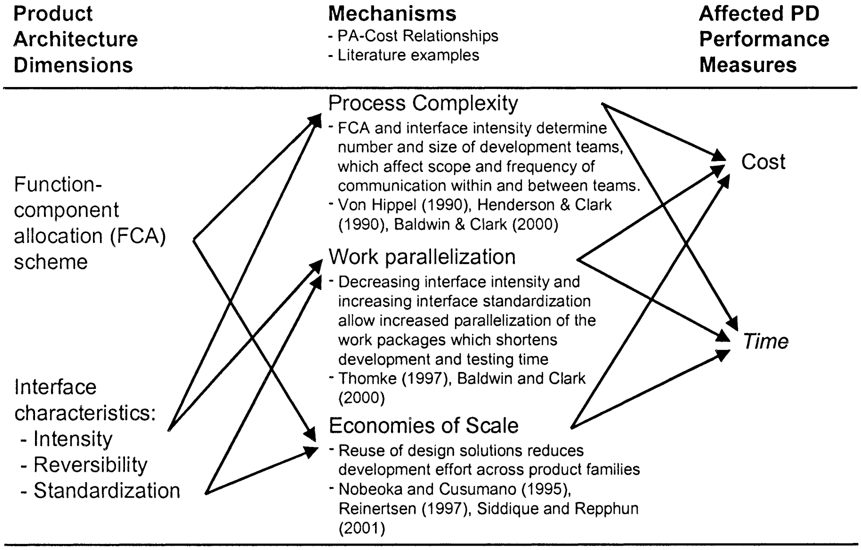


Figure 13-9. Effects of product architecture characteristics on development costs and time.

2.2.3 Product architecture effects on costs of the production phase

With respect to the impact of product architecture decisions on costs that occur during the production phase two sub-sets of processes require separate discussions: (1) manufacturing and assembly, and (2) logistics.

To understand how the first dimension of product architecture, i.e., the size and number of components (function-component allocation scheme) affects manufacturing and assembly costs it is helpful to review the basic idea behind design-for-manufacturing (DFM) and design-for-assembly (DFA) guidelines. Both guidelines help the designer to focus on product characteristics that consume avoidable resources during manufacturing and assembly, respectively, but each with a different rationale. DFM aims at simplifying manufacturing processes, which results—in addition to lower investment—in reduction of process variability and ultimately in faster process rates and higher yields, and thus lower cost. In contrast, DFA generally emphasizes part count reduction, the use of only one assembly direction and the preference of symmetrical parts (Boothroyd, et al., 2002). Empirical evidence exists that supports both claims individually. In case of automobile rear lamp production, for instance, it has been found that complex products requiring complex manufacturing processes result in higher costs compared to simpler parts producible with simpler processes (Banker, et al., 1990). On the other hand, in an analysis of the costs of electromechanical assemblies it has been found that the assembly cost savings through part count reductions can be significant (Boer and Logendran, 1999). Part count reduction is generally seen as a cost reduction tool (Schonberger, 1986; Galsworth, 1994). These findings result in cost curves that increase in opposite directions with respect to the optimal number, and thus complexity, of modules into which a product should be decomposed. The minimum of the sum of the two curves depends on their specific shapes (see Figure 13-10).

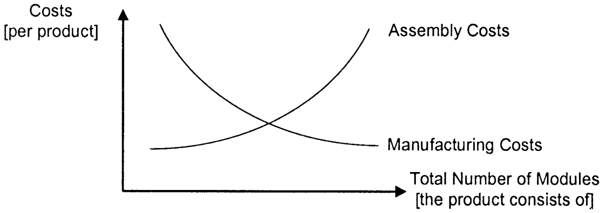


Figure 13-10. Manufacturing and assembly cost behavior with respect to number of modules.

The argument for products requiring simpler manufacturing processes rests essentially on the idea that these processes perform faster and more reliably than their more complex counterparts. Assuming that simpler products require simpler manufacturing processes, this means the product feature complexity affects the efficiency of the process, which in turn directly affects the costs via process speeds and yields. In other words, a design that allows processes to be robust is more likely to consume fewer resources. With respect to product architecture, this observation means that the designer should strive to keep the size of modules or chunks below a complexity level that makes them difficult to manufacture. On the other hand, the argument for products requiring fewer parts (and, as a consequence, fewer manufacturing processes and assembly steps) to achieve lower costs is immediately obvious, as long as the reduction of the number of processes is not paid for with lower yields in the remaining ones. A shift from one manufacturing process to another to reduce part count can have a dramatic impact on assembly time and cost. For example, the instrument panel for the cockpit of the commercial aircraft Boeing 767-4ER used to be manufactured from 296 sheet metal parts and assembled with 600 rivets. A move to precision casting has reduced the part count to 11 and the assembly time from previously 180 hours to 20 hours (Vollrath, 2001). In sum, the product's function-component allocation, i.e., its number and size (complexity) of components, affects both manufacturing and assembly costs, typically in opposing directions, and designers need to develop an understanding of the relative importance of these cost elements in their particular environment.

From a unit cost perspective there is one other effect of product architecture on production costs: this is the use of common components across product families. If the fixed cost portion of manufacturing and assembly can be distributed across a larger number of units, the unit production costs decrease. However, the magnitude of these savings needs to be compared with the potential cost penalties for over-designing a sub-unit or module. For example, products whose costs are dominated by materials costs, i.e., variable costs, such as automotive wire harnesses, may not gain much through the use of commonality (Thonemann and Brandeau, 2000). More generally, the resource use-rate typically decreases with component commonality, but the cost-rate (per cost driver) often increases (Labro, 2004); the final outcome depends on the specific circumstances.

In addition to the product architecture characteristic number and complexity of chunks, the characteristics of the interfaces between the chunks influence the production costs. Interfaces preferred from the low cost production perspective are such that they minimize complexity and uncertainty within the production process. This means, the better the process is known and the more likely it can be performed successfully, and the lower the total number of different processes in the production system is, the lower the expected production costs. The nature and intensity of the interfaces can also be relevant to the production. For example, electronic interfaces consisting of only a plug and a socket may be easier to assemble error-free than a complex mechanical rod connection.

The second subset of production costs is concerned with the aspects of logistics. For the purpose of this chapter, logistics costs encompass costs for storage, transportation, inventory, and work-in-process (WIP). Storage and transportation need to be considered between suppliers and plant, inside the plant, and between plant and customers. Product architecture decisions—the specification of the product's function-component allocation and its interfaces—are most likely to affect these costs to the extent to which they determine packing space and product protection requirements.

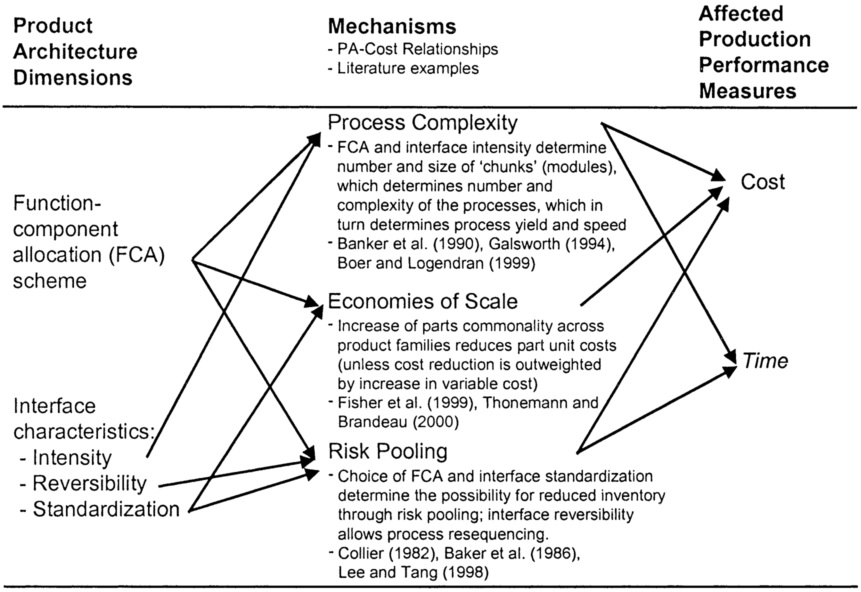


Figure 13-11. Effects of product architecture characteristics on production costs and time.

Product architecture differences also impact the costs for inventory and WIP. The more a product architecture allows late customization or postponement strategies, the more it can contribute to savings in storage and WIP costs through pooling effects. Parts commonality has been identified as a way to reduce the safety stock level for a given service level (Collier, 1982). Others have shown, however, that while the stock for a common part can be lower compared to the unique parts it replaces, the safety stock of the remaining unique components increases if a certain service level is to be maintained (Baker, et al., 1986). These findings have been confirmed for an arbitrary number of products and joint distribution as long as the costs for the product-specific components (that are replaced by a common one) are the same (Gerchak, et al., 1988). For the two-product case, cost ratios have been derived that bound the advantage of the use of common components (Eynan and Rosenblatt, 1996). Another strategy to reduce inventory is to move the common inventory as much upstream in the supply chain as possible to wait with the product customization as much as possible. This strategy might require a re-sequencing of the operations (Lee and Tang, 1998). The key product architecture characteristic for this strategy is the interface reversibility. If it is low, an operation reversal may not be possible because the technical nature of the operations prohibits a reversal (e.g., in the case of steel components welding has be completed before painting). In sum, the use of common parts can reduce inventory, but it needs to be investigated with the specific demand pattern, the relative costs of the components, and other product architecture constraints in mind.

The product architecture's effect on time can have an additional impact on costs via the detour of increasing demand volatility. Because demand volatility increases upstream ('bullwhip-effect'), product architectures can reduce this effect if they allow for parallelization of production to achieve short lead times. Long lead times, together with high levels of demand uncertainty, can amplify the bullwhip-effect and create significant additional costs in the supply chain (Levy, 1994). Overall, a complete assessment of the impact of architectural characteristics on production costs should incorporate manufacturing, assembly, and logistics costs, and evaluate how to balance these different effects. Figure 13-11 summarizes the effects of individual product architecture characteristics on production cost and time.

2.2.4 Product architecture effects on costs of the use phase

In general, three types of costs occur during product use: (1) the costs for operation, (2) the costs for maintenance, and (3) all external costs incurred by the operation of the product.

Most products require some input to operate them. The costs for these inputs can be for fuel or utilities like energy, water, or pressurized air, or costs incurred by the product's characteristic, for instance labor requirements for a machine operation. While it is very difficult to make a general statement about the relationship between product architecture characteristics and operation costs, some issues can be pointed out. Operating costs typically contain two types: costs for standard operation and costs for preparation activities, for example training. The training of personnel is analogous to the setup of a machine: a process necessary to begin operation. Similar to the production arena, if the set-up time, i.e., training time, can be reduced, then the system's productivity increases. A product architecture can contribute to this reduction in 'set-up time' by utilizing common components across members of a product family (which requires proper function-component alignment). For example, aircraft producers are trying to install similar, if not identical, cockpits into airplanes of different sizes to reduce the airlines' need to retrain their crews (Anonymous, 2005). Similarly, if it is not the operator that changes (as in the airplane case) but the task that the product has to accomplish (e.g., a machine tool that is planned to produce a variety of components) then a product architecture that supports to reconfigure the product quickly is advantageous (Landers, et al., 2001). Proper function-component alignment and high degrees of interface reversibility are key in this situation to improve the productivity of the product by reducing its set-up costs and, thus, its operating costs as measured by units produced per time unit.

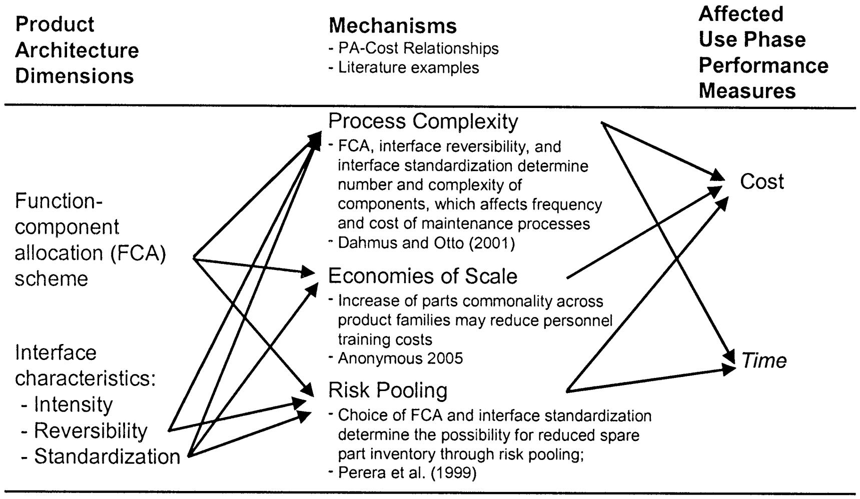


Figure 13-12. Effects of product architecture characteristics on use phase costs and time.

With respect to maintenance costs that occur during a product's use phase two major questions are relevant. First, what is the likelihood that maintenance (and its costs) will occur during the product's use phase and, second, what will be the anticipated costs for this maintenance procedure? Grouping parts with similar expected lifetimes together is likely to reduce the repair and replacement costs by minimizing the required parts replacement processes (Dahmus and Otto, 2001). A proper module definition (function-component allocation) can help achieving this goal. In addition, a product architecture that allows easy and fast access for maintenance and repair requires less time to execute the actual maintenance procedure and, consequently, leads to lower maintenance costs. The product architecture characteristic interface reversibility is the important design variable in this case. Also, in case that a product has multiple identical parts (function component allocation) fewer parts need to be stocked in inventory (compared to unique parts) for providing the same level of availability (Perera, et al., 1999). Like risk pooling across products in production, this strategy translates into lower spare part inventory costs as part of the maintenance costs. Note that the different elements of maintenance costs described above may react differently to the same product architecture design decision.

Finally, the operation of any product may also cause so-called external costs, e.g., damages to public health or the environment through emissions. A link between product architecture design decisions and external costs is very difficult, if at all, to establish and goes beyond the scope of this work. Figure 13-12 recapitulates the effects of individual product architecture characteristics on costs and time in the product use phase.

2.2.5 Product architecture effects on costs of the retirement phase

In the last phase of a product's life cycle, costs are created through activities like disassembly or disposal. In addition to these direct costs, external costs, like degradation of the environment or air quality, can occur.

To estimate disassembly costs as a function of the product architecture is very difficult, particularly since it is often unclear which disassembly sequences is the most economically viable one. The reverse of the assembly process may, or may not, be the most cost effective way to disassemble the product. Researchers have suggested a number of scoring processes to compare disassembly efforts for different designs. Some suggest comparing disassembly costs for different designs on a relatively high level of aggregation. Emblemsvag and Bras (1994), for instance, propose to list all activities the disassembly of various products would require, compute the costs for each activity per time unit, determine the time each design requires each activity, and compare the results. This type of analysis, however, does not reveal specifically which architectural features make one design more costly to disassemble than another. To answer this type of question more detailed analyses are required. Das, et al. (2000), for example, propose to compute a disassembly effort index based on seven factors, such as time, tools, fixtures, access, instruct, hazard, and force requirements. The fact that both the score for each of these factors as well as the weights among them are based on qualitative assessments demonstrates the difficult nature of the task to estimate disassembly costs unambiguously. Others have extended this work to include bulk recycling in addition to disassembly activities (Sodhi and Knight, 1998). However, while the product architecture affects disassembly costs (via the dimensions function-component allocation scheme and interface reversibility), its impact on bulk recycling is only relevant together with the specific values of the materials involved. Finally, while determining the costs to landfill a product (or parts of it) is relatively straightforward, the results, however, are unlikely to depend on architectural characteristics of the product (leaving material consideration aside). Figure 13-13 summarizes the product architecture's effects on costs in the product retirement phase.

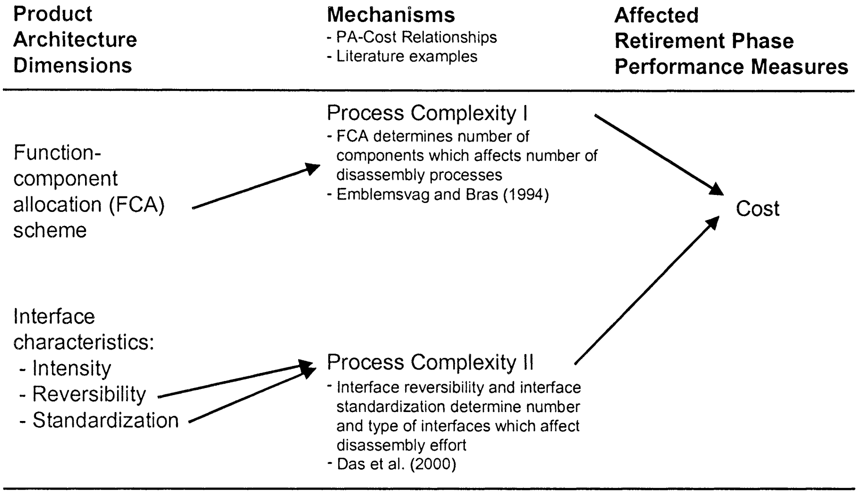


Figure 13-13. Effects of product architecture characteristics on retirement phase costs.

This section has demonstrated that the designer responsible for defining the product's architecture faces a difficult task. Since the analysis for product architecture costing requires a decision on which life cycle phase to include, the designer must develop an understanding of the longitudinal tradeoffs that product architecture design decisions face between life cycle phase and within individual life cycle phases. The second step of the roadmap presented in this section provides a guideline to develop this understanding.

2.3 Step 3: Determine relevant cost allocation rules and their impact on cost analyses

Once the various cost types that can occur over a product's life and the relationships between product architecture design decisions and these costs are identified, the third step of the roadmap requires to determine the rules for the cost allocation procedures. Particularly relevant for the results of any cost analysis are the—often only implicit—assumptions on the analysis boundaries, on the overhead allocation mechanisms, and on the dynamics of the process under investigation.

2.3.1 Unit of analysis

Typically, product unit costs are chosen for cost comparisons of assembled products. There are, however, other units of analysis that could be selected alternatively: product families, product programs, departments, factories, companies, or entire economies. The order of this list of potential levels of analysis indicates an increasing distance from the physical object itself. While a cost analysis focusing on a product makes it easy to assess costs that are directly related to the product (e.g., material consumption), it makes the allocation of more 'distant' costs (e.g., factory guards) very difficult. On the other hand, for cost analyses on a company level, almost all costs are somewhat 'direct' (see Figure 13-14).

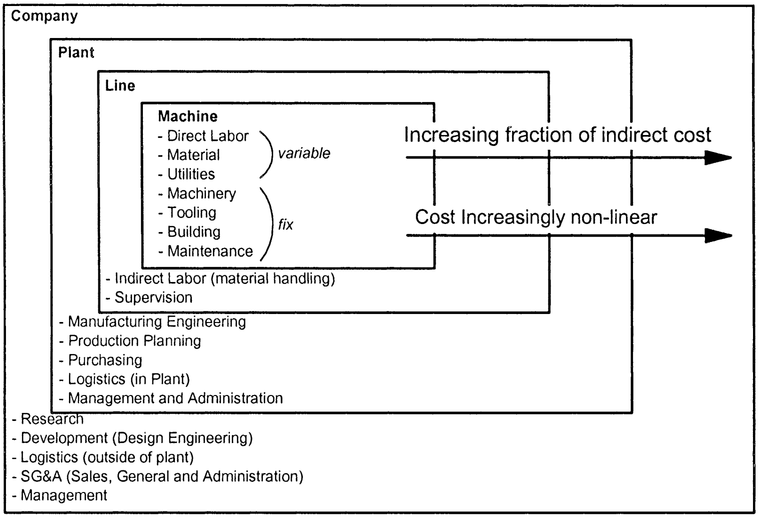


Figure 13-14. Different levels of cost analysis.

The direct-indirect cost classification depends on the choice of the cost object. "A useful rule of thumb is that the broader the definition of the cost object, the higher the proportion of its total costs are its direct costs—and the more confidence management has in the accuracy of the resulting cost amounts. The narrower the definition of the cost object, the lower the proportion of its total costs are its direct costs—and the less confidence management has in the accuracy of resulting cost amounts." (Horngren and Foster, 1991:28) Since product architecture costing is concerned with the cost effects that product architecture choices trigger, it is logical to focus the cost analysis on a level where product architectures can be distinguished, i.e., on the product or product family level. This in turn creates the above mentioned allocation problem of how to allocate the significant indirect cost portion, often called 'overhead.' Overhead usually encompasses costs with various levels of 'indirectness.' For the interpretation of cost consequences of product architecture design decisions it is very important to understand the mechanisms by which these overhead costs are allocated.

2.3.2 Allocation of overhead costs

The accounting literature employs two distinctions for costs: direct versus indirect costs and fixed versus variable costs (Horngren and Foster, 1991). While the first uses the cost traceability to separate direct from indirect costs, the second uses the dependency with regards to changes in production volume as a measure to classify fixed and variable costs.

In the production arena, costs that are typically considered variable are costs for direct labor, materials, and utilities. In contrast, machinery, tooling, and building costs are usually considered fixed costs. These distinctions, however, are not clear-cut, but depend on the chosen time horizon, the chosen manufacturing technology, and the chosen accounting principle. A change in the chosen time horizon can turn the same costs from fixed into variable costs. Labor costs are typically viewed in short time frames as fixed costs whereas in the long run they are typically treated as variable in nature. The choice of a manufacturing technology may determine whether a specific or a generic tool is deployed. A shear as a cutting tool that can be used to produce other products as well exhibits variable cost behavior whereas a specific cutting die that does the same job, but can only be used for this specific product becomes fixed costs. Finally, certain accounting principles can shift costs from the fixed costs category into the variable cost category, and vice versa. The assumption, for example, that free machine capacity can be employed for other jobs turns the allocated machine costs effectively into variable costs whereas the assumption that the machine is dedicated to a specific product results in fixed cost behavior.

In sum, what is typically called overhead is a broad category with often fuzzy boundaries. It is, however, a category that becomes increasingly important due to increasing product and process complexity, shrinking direct labor content, shorter product life cycles, and increasingly heterogeneous markets (Miller and Vollmann, 1985; Doran and Dowd, 1999; Cokins, 2000). Table 13-1 gives an overview of the magnitude of some overhead costs found in recent studies.

One characteristic feature of overhead costs is their lack of direct dependency on production volume. Activities that support in various ways the actual production processes do not necessarily vary in direct proportionality with the production volume. It has been argued that the costs for these activities vary with the intensity or frequency of these activities. For example, the time and manpower to write a purchasing order does not vary with the number of equal parts ordered, but each order incurs an average cost for the transaction 'write purchasing order.' This insight triggered the development of activity-based costing (ABC) (Kaplan, 1991; Kaplan and Cooper, 1998). ABC promotes a cost allocation process in proportion to the activities consumed by the products produced. The basic idea of ABC is to calculate the costs of activities (cost drivers) and 'charge' products with the time with which they consume an activity times the use rate per time unit. The cost drivers can be on various levels in the firm: "While some activity cost drivers are unit-related (such as machine and labor hours), as conventionally assumed, many activity cost drivers are batch-related, product-sustaining, and customer-sustaining" (Cooper and Kaplan, 1992:4).[20]

Table 13-1. Overhead costs found in recent studies.

Author(s)	**Total Costs (=100%)**			**Activities considered**	**Industry**
	Direct Material	**Direct Labor**	**Mfg. Overhead (MO)**		
(Banker, et al., 1995)	65.4%	8.9%	25.7%	Plant level study	Electronics, Machinery, Automobile components (mean values of 32 facilities)
(Foster and Gupta, 1990)	54.3%	6.6%	39.1%	Procurement, Production, Support	Electronics (mean values of 37 facilities)
(Galsworth, 1994: 85)	40%-65%		35%-60%	Total Costs: Function cost: 40% Variety cost:25% Control cost: 35%	Manufacturing
(Hundal, 1997)	45%-65%	8%-20%	22%-40%	Not specified	Aerospace, Computers, Electronics, General Equip., Automobiles
(Miller and Vollmann, 1985)	20%-40%		60%-80%	Overhead Costs: G & A 20% Indirect Labor 12% Engineering 15% Equipment 20% Materials OH 33%	Electronics

While ABC represents an invaluable step towards a better understanding of how to allocate what used to be called 'overhead,' it is still helpful to review some of the assumptions that underlie even ABC with respect to product variety. More specifically, these assumptions are concerned with linearity of activity-cost relationships, with different types of variety, and sequence-dependent variety costs.

---

[20] Some have criticized ABC as leading to poor short-term decisions, and suggested the Theory-of-Constraints (TOC) as a better tool for short-run cost allocations. TOC assumes all costs other than direct material as fixed (Goldratt and Cox, 1984). Then, to maximize profitability, TOC seeks to maximize throughput. TOC promotes finding the bottleneck in an existing system and adjusting all other production to it to eliminate inventory. In the debate about whether ABC or TOC is the superior way of interpreting costs, various authors argue to understand both methods as opposing ends of a continuum with respect to planning time horizon: ABC for long-range planning, TOC for short-term decisions (Fritzsch, 1997; Cooper and Kaplan, 1998; Kee, 1998). Since the choice of the product architecture is a rather long-term decision, ABC is the more relevant method for our purposes here.

ABC argues that many overhead costs are related to activity type and activity frequency rather than production volume. Standard ABC typically assumes a linear relationship between activity and cost. The limits of this assumption, however, become apparent in case of product variety. Product variety often causes additional work in activities such as planning, control, monitoring, and coordination (Lingnau, 1999). Not only does this cost propagation effect make it more difficult to trace individual costs, it often creates also an additional allocation problem: if product variety creates costs above the sum of the costs of the individual products, how are these variety-related extra costs allocated to the individual products?

To make matters more complicated, product variety can also take on different forms each of which has a different effect on costs. For example, Ittner and MacDuffie (1994) defined three levels of product variety in their study of overhead costs in automotive assembly plants: core or fundamental variety (model mix complexity), intermediate variety (parts complexity), and peripheral variety (option complexity). They find empirical support only for the latter two affecting overhead costs, ".. reflecting the considerable logistical, coordination, and supervisory challenges that accompany an increase number of parts and more complex manufacturing tasks." (Ittner and MacDuffie, 1994:29) Another approach to specify product variety has been followed by Anderson (1995) who measures the impact of product mix heterogeneity on manufacturing overhead costs by identifying seven independent product attributes, using engineering specifications. By measuring on the attribute-level, Anderson finds that increased overhead cost "is associated with increases in the number and severity of setups and increased heterogeneity in process specifications (expected downtime) and quality standards (defect tolerance heterogeneity) of a plant's product mix" (Anderson, 1995:383).

Finally, how product variety is distributed over time can affect the effort to balance and sequence a production line. For example, taking a production perspective a study of product variety finds that "[o]ption variability has significantly greater negative impact on productivity than option content in automobile assembly" (Fisher and Ittner, 1999:785). In this case, variety's impact on indirect and overhead labor is much greater than it is on direct labor. The authors explain this with the built-in slack in automotive assembly lines that allows handling option variation in the first place. They point out that because these costs are born through the variability complexity it is difficult to allocate these excess costs to any specific product.

With respect to the question of how the link between product architecture characteristic and cost is influenced by the cost allocation procedure some general observations can be made. A product architecture that allows operations conducted closer on a per-unit basis allows more precise cost allocation. For example, a process that produces only one part at a time allows easy allocation of all non-direct costs (setup, purchasing, etc.). In contrast, product architectures that cause complex logistical, balancing, sequencing, or quality processes may make the cost allocation more difficult. Within limits, these arguments call for products with architectures consisting of fewer, more modular-like components (dimension function-component allocation) and with high levels of interface standardization.

2.3.3 Process dynamics

The third issue of the roadmap's third step: 'determination of cost allocation rules,' is concerned with the extent to which the processes under consideration are considered static or dynamic. There are two cases of non-static situations: (i) a one-time change followed by a static period, and (ii) a change over longer periods of time. In the first case, the relevant issue is the ratio of 'ramp-up period' to 'normal production period.' If, for example, an entire production run will extend over several years and the ramp-up takes only a few days, the cost analysis focus can be put on the system costs assuming it in its static condition. In contrast, if the production run is relatively short and the ramp-up takes up a significant portion of it, the systems costs are not well represented by the production run alone. In some production environments the ramp-up time can represent a significant fraction of total production time, e.g., it can take up to six months to bring an automotive assembly plant up to full production load (Almgren, 2000).

Cost changes over longer periods of time can occur in two ways: the change itself can either be constant or variable. The case in which the change is (for the most part) constant is often caused by what has come to be known as the learning curve effect. The argument is that with accumulating production volume workers and engineers are getting better in what they are doing. They improve the processes and their work environment in a manner that continuously improves their overall productivity. Often times the learning effect is measured as a constant fraction of cost reduction, e.g., 20%, with every cumulative doubling of the production volume. Empirical evidence has been presented that this effect indeed exists (Anderson, 1995). Activity-based costing systems can help to detect these learning effects (Andrade, et al., 1999).

In the second case of changing unit costs the change itself is dynamic, i.e., unit costs do not change by a constant rate but follow dynamic patterns. An example of this phenomenon is the case of non-constant unit costs as a result of different ways of sequencing different products through jointly used production processes. Flexible manufacturing systems (FMS), for example, can manufacture different products on the same machine. The set-up time, however, may depend on what product has been produced prior to the one under consideration. Will the same tool be used? If not, is the tool change time dependent on what tool was used for the previous product? This problem has been addressed through the use of ABC systems in conjunction with production planning models (Koltai, et al., 2000).

With respect to the effects of product architecture choice on unit cost, the phenomenon described in this section cannot be determined with product architecture data alone, but requires data (or assumptions) on the production environment including scheduling and the production program information.

2.4 Step 4: Select cost model appropriate for the design decision at hand

The fourth step of the roadmap for product architecture costing requires the selection of one or several cost models that are appropriate for the design decision at hand. A number of cost models have been developed to help designers to assess the economic consequences of design decisions. The existing models can be grouped into three categories: parametric, analogous, and analytical. Parametric models aim at establishing scaling factors of cost drivers found through analysis of historical data. Regression analysis is a typical method to extract such scaling factors. Due to the simplicity in use, parametric techniques are used in many industries (Bielefeld and Rucklos, 1992; Uppal 1996). Non-parametric methods such as neural networks have also been applied to find design variable-cost relationships (Bode, 2000).

The underlying idea for analogous models is to search for similarities between the design at hand and a large number of historical cases stored in databases. To be able to compare products on multiple levels (product, subassembly, part, etc.) hierarchically structured approaches have been developed (Liebers and Kals, 1997; Rehman and Guenov, 1998; Ten Brinke, et al., 2000). Other approaches focus more on abstract elements like features (Brimson, 1998; Leibl, et al., 1999).

Finally, cost models in the third category, analytical cost models, come in two very different flavors. One category is represented by abstract mathematical models, often used to generate insights into general questions. Their emphasis is mostly on structural tradeoff modeling, while the functions of relationships between individual design decisions and costs are typically assumed to be known in their shape (Roemer, et al., 2000; Thonemann and Brandeau, 2000; Krishnan and Gupta, 2001). The other flavor of analytical models is represented by detailed technical cost models of (mostly) manufacturing processes to estimate the associated costs (Clark, et al., 1997; Locascio 1999; Locascio, 2000; Kirchain, 2001). Technical cost models model manufacturing processes based on the process physics and establish links between a few critical design parameters and the process dynamics, which in turn determine the costs. Existing cost modeling techniques are not discussed here in detail; the interested reader is referred to recent reviews in (Asiedu and Gu, 1998; Layer, et al., 2002).

Instead of presenting the different cost models in detail, this section presents four criteria to help thinking about making the appropriate cost model selection when assessing cost implications of product architecture design decisions. First, does the cost model or technique require a substantial data set of similar cases? Regression analyses or neural networks, for example, usually require sufficient cases to be able to produce relevant cost predictions. Second, how large is the number of acceptable cost drivers? Most cost modeling techniques allow only a limited number of cost drivers. To some extent this question is related to the previous one in that the number of available cases restricts the number of acceptable cost drivers. Third, how large are the acceptable differences between the product architecture candidates under investigation? This criterion is particularly relevant if substantially different product architectures are to be analyzed. Modeling techniques that build on a set of known cases are usually limited when applied to entirely new cases. Finally, what certainty level is required for the input data? As indicated earlier, cost analyses in early design stages typically lack detailed and accurate product design date. The assessment of the cost models along this fourth criterion reveals the underlying modeling philosophy. Some models use search procedures to find relevant data among existing cases (e.g., analogous models) whereas others build the cost analysis for every case anew (e.g., process-based cost models). Depending on the goal of the product architecture analysis and the available data, different methods are advantageous. Table 13-2 summarizes the various cost modeling approaches with respect to product architecture costing.

Table 13-2. Assessment of various cost estimation models along four application criteria.

Cost Estimation Models	**Application Criteria**	**Data Set Requirement (min case base)**	**Acceptable Number of Cost Drivers**	**Acceptable Difference in Architecture Decomposition**	**Required Certainty of Data Input**
Parametric	Regression Analysis	Large	Low	Small	Medium
	Complexity-Theory Based	Medium	Low	Small	High
	Neural Networks	Large	Low	Small	Medium
Analogous	Feature-Based	Medium	Low	Small	High
	Expert Systems	Large	Medium	Medium	High
Analytical	Abstract Modeling	Small	Small	Small	None
	Process-Based Cost Models	Small	Medium	Large	Medium

3. CONCLUDING REMARKS

This chapter has introduced a roadmap for product architecture costing. Each step of the roadmap prior to the actual modeling of a specific situation, i.e., (1) to assess differences of product architecture candidates, (2) to identify relevant product life cycle phase(s) and their product architecture-cost relationships, (3) to determine relevant cost allocation rules, and (4) to select cost models appropriate to the situation at hand have been discussed in detail. This comprehensive discussion of how individual product architecture characteristics affect specific cost elements over a product's life cycle can serve as a guideline when formulating various tradeoffs. For example, a manufacturer of long-living products, e.g., a ship builder, might want to tradeoff costs for building the ship with the costs for operating it. In contrast, a manufacturer of mass-produced consumer goods might be more interested in the cost tradeoff between the costs for parts fabrication and the costs for assembly. For any given firm, the determination of the relevant tradeoffs is impacted by such factors as the firm's business model, its warranty policies, and its competitive and legal environment. The roadmap also provides an overview of how cost allocation rules can affect cost analyses results, and thus the cost advantage of one product architecture over another. Finally, the roadmap includes a categorization of existing cost models, and illustrates which one to select depending on the size of the available data set, the given data set's level of variation and accuracy, and the number of acceptable cost drivers.

This roadmap for product architecture attempts to provide a comprehensive consideration of the relevant questions when conducting an analysis of the cost consequences of product architecture differences. The relationships identified and cost models presented can now serve as stepping stones for the development of user-friendly design guidelines as well as of more complex optimization models (see Figure 13-1). Some thoughts on how these next steps could proceed follow.

The development of product architecture design guidelines that lead the designer towards 'better' product architectures, given the requirements that the product faces, can be envisioned similar to the development of the well-known DFM/DFA guidelines. The DFM/DFA guidelines represent the condensed experience across many cases of design changes with respect to manufacturing and assembly. Similarly, a database containing the results of many specific product architecture-cost analyses could be used to search for more general patterns of cost effects that are due to differences in product architecture. As a step in this direction a firm might build a repository of their own cost data and associate the data with the corresponding product architecture characteristics. This way the firm might populate the product architecture-cost space with more of its own data points. Over time, this would offer the chance to introduce internal learning into the product architecture design process (Anderson and Sedatole, 1998) and would foster the construction of product architecture design guidelines.

The product architecture-cost relationships identified by the roadmap can also inform the process of building more complex models that can support the product architecture development process more dynamically. While knowing the cost effects, allocation rules, and cost models discussed in this chapter allows evaluating cost consequences of differences along individual product architecture characteristics, this knowledge does not automatically feed back into the product architecture design process. If it were possible to turn product architecture characteristics into variables that exist across the entire solution space—which they currently often do not—they could be used to find optimal architectures, optimal with respect to the cost determined as relevant. With respect to the product architecture development process, this would replace the process of selecting among product architecture candidates with one that helps designers to develop more cost effective product architectures by giving immediate feedback to product architecture design suggestions. One particularly promising extension of this research direction on product architecture costing is the treatment of uncertainty. While uncertainty is inherent in any estimation of future data, the way in which it is modeled might provide additional insights for the product architecture selection and development decisions. While deterministic cost models can be augmented with sensitivity analyses, more sophisticated measures of risk and uncertainty could advance the cost modeling tools, and by extension, the product architecture creation.

4. ACKNOWLEDGMENTS

For helpful comments and discussions on cost modeling the author would like to thank Joel Clark, Frank Field, Randolph Kirchain, and Rich Roth. In addition, the chapter benefited from helpful comments from three anonymous reviewers. Research funding from the International Motor Vehicle Program (IMVP) at MIT is gratefully acknowledged.