ELSEVIER

Research Policy 24 (1995) 419–440

research policy

The role of product architecture in the manufacturing firm

Karl Ulrich *

Massachusetts Institute of Technology, Sloan School of Management, Room E53-390, Cambridge, MA 02139, USA

Final version received December 1993

Abstract

Product architecture is the scheme by which the function of a product is allocated to physical components. This paper further defines product architecture, provides a typology of product architectures, and articulates the potential linkages between the architecture of the product and five areas of managerial importance: (1) product change; (2) product variety; (3) component standardization; (4) product performance; and (5) product development management. The paper is conceptual and foundational, synthesizing fragments from several different disciplines, including software engineering, design theory, operations management and product development management. The paper is intended to raise awareness of the far-reaching implications of the architecture of the product, to create a vocabulary for discussing and addressing the decisions and issues that are linked to product architecture, and to identify and discuss specific trade-offs associated with the choice of a product architecture.

1. Introduction

Product architecture is the scheme by which the function of a product is allocated to physical components. This paper argues that the architecture of the product can be a key driver of the performance of the manufacturing firm, that firms have substantial latitude in choosing a product architecture, and that the architecture of the product is therefore important in managerial decision making.

Product architecture is particularly relevant to the research and development (R&D) function of a company, because architectural decisions are made during the early phases of the innovation process where the R&D function often plays a lead role. While these architectural decisions are linked to the overall performance of the firm, they are also linked to specific R&D issues, including the ease of product change, the division between internal and external development resources, the ability to achieve certain types of technical product performance, and the way development is managed and organized.

In making these arguments, the paper builds on knowledge from several somewhat disparate research communities: design theory, software engineering, operations management and management of product development. My approach is to synthesize fragments of existing theory and knowledge into a new framework for understanding product architecture, and to use this framework to illuminate, with examples, how the architecture of the product relates to manufacturing firm performance. My intention is that industrial

* Corresponding author.

SSDI 0048-7333(94)00775-O

practitioners will benefit from the argument and develop a stronger conceptual foundation for decision making, and that researchers will benefit from the argument through an enhanced ability to formulate focused research questions around these issues.

The paper consists of eight remaining sections. Section 2 defines product architecture. Section 3 provides a typology of architectures. Sections 4 through 8 articulate the linkages among product architecture, product change, product variety, component standardization, product performance and the management of product development. Finally, Section 9 summarizes the key points, discusses how to establish a product architecture, and identifies three promising research directions.

2. What is product architecture?

In informal terms, the architecture of the product is the scheme by which the function of the product is allocated to physical components. I define product architecture more precisely as: (1) the arrangement of *functional elements*; (2) the mapping from *functional elements* to *physical components*; (3) the specification of the *interfaces* among interacting physical components.

This section expands on this definition using the example of a *trailer* to illustrate the key points.

2.1. The arrangement of functional elements

The function of a product is what it does as opposed to what the physical characteristics of the product are. There have been several attempts in the design theory community to create formal languages for describing function [7], and there have been modest successes in narrow domains of application such as electro- and fluid-mechanical systems and digital circuits [32]. There have also been efforts to create informal functional languages to facilitate the practice of design [19,11]. These languages are frequently used to create diagrams consisting of functional elements, expressed as linguistic terms like ‘convert energy’, connected by links indicating the exchange of signals, materials, forces and energy. Some authors of informal functional languages provide a vocabulary of standard functional elements, while others rely on users to devise their own. Functional elements are sometimes called *functional requirements* [28] or *functives* [8], and the function diagram has been variously called a *function structure* [19,11], a *functional description* and a *schematic description* [32]. Consistent with

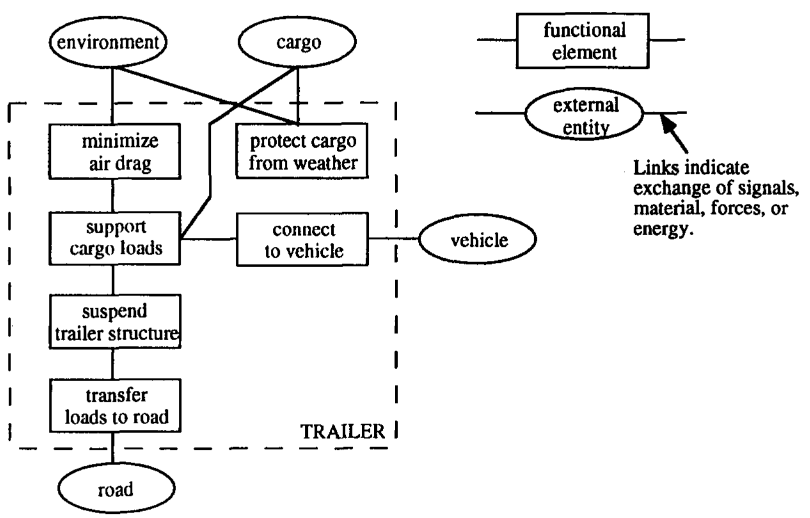

Fig. 1. A function structure for a trailer.

Pahl and Beitz, and Hubka and Eder, I call the arrangement of functional elements and their interconnections, a *function structure*. An example of a function structure for a trailer is shown in Fig. 1.

Function structures can be created at different levels of abstraction [8]. At the most general level, the function structure for a trailer might consist of a single functional element — 'expand cargo capacity'. At a more detailed level, the function structure could be specified as consisting of the collection of functional elements shown in Fig. 1, i.e. *connect to vehicle*, *protect cargo from weather*, *minimize air drag*, *support cargo loads*, *suspend trailer structure*, and *transfer loads to road*.

As they are expressed in more detail, function structures embody more assumptions about the physical working principles on which the product is based. For example, expand cargo capacity does not assume the trailer will be a device towed over the road (the trailer could be a lighter-than-air device), while the more detailed function structure shown in Fig. 1 does embody this assumption. For this reason, two products that at the most general level do the same thing may have different function structures when described at a more detailed level.

While most functional elements involve the exchange of signals, materials, forces and energy, some elements do not interact with other functional elements. An example of such an element might be *harmonize aesthetically with vehicle*.

2.2. The mapping from functional elements to physical components

The second part of the product architecture is the mapping from functional elements to physical components. A discrete physical product consists of one or more components. For clarity, I define a component as a separable physical part or subassembly. However, for many of the arguments in the paper, a component can be thought of as any distinct region of the product, allowing the inclusion of a software subroutine in the definition of a component. Similarly, distinct regions of an integrated circuit, although not actually separate physical parts, could be thought of as components.

Physical components implement the functional elements of the product. The mapping between functional elements and components may be one-to-one, many-to-one, or one-to-many. Two different trailer designs and their associated mappings of functional elements to components are shown in Figs. 2 and 3.

2.3. The specification of the interfaces between interacting physical components

By definition, interacting components are connected by some physical interface. Interfaces may involve geometric connections between two components, as with a gear on a shaft, or may involve non-contact interactions, as with the infrared communication link between a remote control

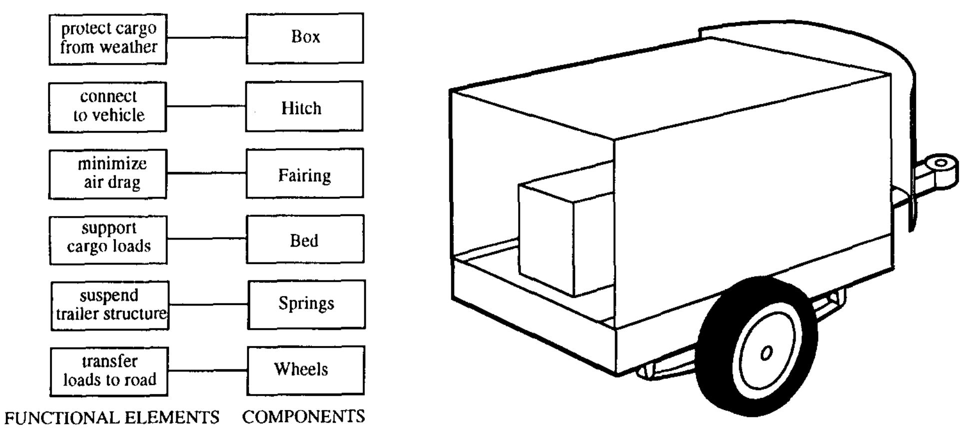

Fig. 2. A modular trailer architecture exhibiting a one-to-one mapping from functional elements to physical components.

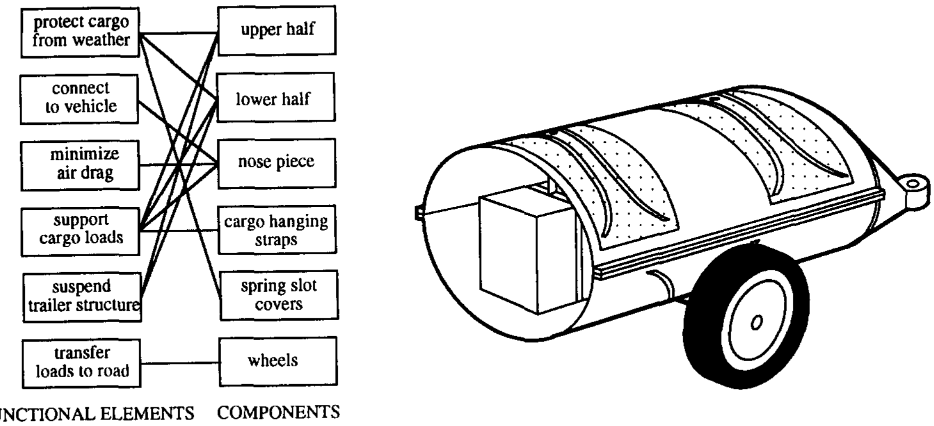

Fig. 3. An integral trailer architecture exhibiting a complex mapping from functional elements to physical components. (The upper and lower halves of the trailer have slots cut in them. The strip of material remaining between two slots acts as a leaf spring. The cargo is hung by straps from the two springs in the upper half. The axle is attached to the spring in the lower half. Covers, shown shaded, are attached over the slots. The nose piece is the component containing the trailer hitch.)

and a television set. An interface specification defines the protocol for the primary interactions across the component interfaces, and the mating geometry in cases where there is a geometric connection.

For example, one of the interfaces for the trailer shown in Fig. 2 is between the box and the bed. The specification of the interface includes the dimensions of the contact surfaces between the two components, the positions and sizes of the bolt holes, and the maximum force the interface is expected to sustain.

Note that interfaces may be specified to adhere to a standard protocol. Examples of protocols that have been standardized across many different manufacturers' products are: SCSI (small computer systems interface), tyre/rim standards for automobiles, a stereo 'phono' jack, a garden hose connection thread and a 'ball-type' trailer hitch. Manufacturers sometimes choose to adopt a common protocol for interfaces used within their own product line, even though the interface may not adhere to an external standard.

3. A typology of product architectures

A typology of architectures provides a vocabulary for discussing the implications of the choice of architecture on the performance of the manufacturing firm. The first distinction in the typology is between a *modular* architecture and an *integral* architecture. A modular architecture includes a one-to-one mapping from functional elements in the function structure to the physical components of the product, and specifies de-coupled interfaces between components. An integral architecture includes a complex (non one-to-one) mapping from functional elements to physical components and/or coupled interfaces between components.

3.1. Types of mappings from functional elements to physical components

The two trailers in Figs. 2 and 3 illustrate two extreme examples of mappings from functional elements to components. One trailer embodies a one-to-one mapping between functional elements and components. Assuming that the component interfaces are de-coupled (more on this later), this trailer has a modular architecture. In the field of software engineering, the notion of module *cohesion* or *strength* is similar to the one-to-one mapping of functional elements to components [25]. The other trailer embodies a mapping in which several functional elements are each implemented by more than one component, and

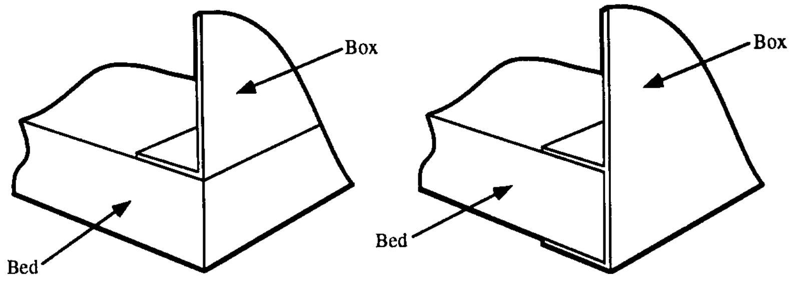

Fig. 4. Two example interfaces between the trailer box and trailer bed; one de-coupled, the other coupled. The coupled interface requires that the box be changed whenever a change in the thickness of the bed is made to accommodate increased structural loading.

in which several components each implement more than one functional element (a complex mapping). This trailer has an integral architecture. The phenomenon of a single component implementing several functional elements is called *function sharing* in the design theory community and is described in detail by Ulrich and Seering [33].

To some extent, whether or not functional elements map to more than one component depends on the level of detail at which the components and functional elements are considered. For example, if every washer, screw and filament of wire is considered a component, then each functional element will map to many components. In order to more precisely define what a one-to-one mapping between functional elements and components means, consider a product disassembled to the level of individual piece parts. (This level of disassembly has been called the *iota* level[1].) In general, many possible subassemblies [2] could be created from these iota parts. If there is a partitioning of the set of iota parts into subassemblies such that there is a one-to-one mapping between these subassemblies and functional elements, then the product exhibits the one-to-one mapping characteristic of a modular architecture.

3.2. Interface coupling

In addition to one-to-one mappings, modular architectures include de-coupled component interfaces. Two components are coupled if a change made to one component requires a change to the other component in order for the overall product to work correctly. Two physical components connected by an interface are almost always coupled to some extent; there is almost always a change that can be made to one component that will require a change to the other component. (For example, arbitrarily increasing the operating temperature of one component by 1000°C will require a change to nearly any imaginable neighbouring component.) However, in practical terms, coupling is relevant only to changes that modify the component in some useful way. (See [25] for a detailed discussion of the different types of coupling encountered in software.)

Fig. 4 illustrates an example of an interface between two components, the bed and the box from the trailer in Fig. 2. The coupled interface embodies a dependency between the thickness of the bed and the vertical gap in the box connection slot. The de-coupled interface involves no such dependency. For the coupled interface, when

[1] I have seen this term used at the General Motors Vehicle Assessment Center to describe the parts resulting from a complete disassembly of a vehicle, down to the last nut, bolt and washer.

[2] A subassembly is a collection of components that: (1) can be assembled into a unit, and (2) can be subsequently treated as a single component during further assembly of the product.

the thickness of the bed must be changed to accommodate a change in the cargo load rating, the box must change as well. Although the example in Fig. 4 is geometric, coupling may also be based on other physical phenomena, such as heat or magnetism.

3.3. *Types of modular architectures*

I divide modular architectures into three subtypes: *slot*, *bus* and *sectional*. Because each of the three sub-types is modular, each embodies a one-to-one mapping between functional elements and components, and the component interfaces are de-coupled; the differences among these sub-types lie in the way the component interactions are organized.

3.3.1. *Slot*

Each of the interfaces between components in a slot architecture is of a different type from the others, so that the various components in the product cannot be interchanged. An automobile radio is an example of a component in a slot architecture. The radio implements exactly one function and is de-coupled from surrounding components, but its interface is different from any of the other components in the vehicle (e.g. radios and speedometers have different types of interfaces to the instrument panel).

3.3.2. *Bus*

In a bus architecture, there is a common bus to which the other physical components connect via the same type of interface. A common example of a component in a bus architecture would be an expansion card for a personal computer. Non-electronic products can also be built around a bus architecture. Track lighting, shelving systems with rails and adjustable roof racks for automobiles all embody a bus architecture. The bus is not necessarily linear; I also include components connected by a multi-dimensional network in the bus subtype.

3.3.3. *Sectional*

In a sectional architecture, all interfaces are of the same type and there is no single element to which all the other components attach. The assembly is built up by connecting the components to each other via identical interfaces. Many piping systems adhere to a sectional architecture, as do sectional sofas, office partitions and some computer systems.

Figs. 5–7 illustrate this typology for the trailer example, for a desk, and for a personal computer. I intend for the typology to provide a vocabulary for describing different product architectures. The types shown are idealized; most real products exhibit some combination of the characteristics of several types. Products may also exhibit characteristics of different types depending on whether one observes the product at the level of the overall final assembly or at the level of individual piece parts and subassemblies.

A firm can design and manufacture products without ever explicitly creating a product architecture or even a function structure. In the domains of software and electronic systems, the idea of a function structure (labelled as a *schematic*, *flow chart*, etc.) is prevalent in industrial practice [17,25]. However, the notion of a function structure is just beginning to be disseminated in many mechanical domains. (See for example Ullman [30] for a recent mechanical design textbook adopting the idea.) If a product architecture is explicitly established during the product development process, this step usually occurs during the system-level design phase of the process after the basic technological working principles have been established, but before the design of components and subsystems has begun.

The examples in Figs. 5–7 suggest that firms possess substantial latitude in choosing a product architecture, although the architecture of many existing products may be less the result of deliberate choice and more the result of incremental evolution. Several scholars have prescribed a modular architecture as ideal. For example, Suh [28] argues that a modular architecture is an axiom of good design, and Alexander [1] presents an 'optimal' design methodology, ensuring a lack of coupling between components. (Although neither author argues his point in my terminology.) I maintain that while product architecture is extremely important, no single architecture is opti-

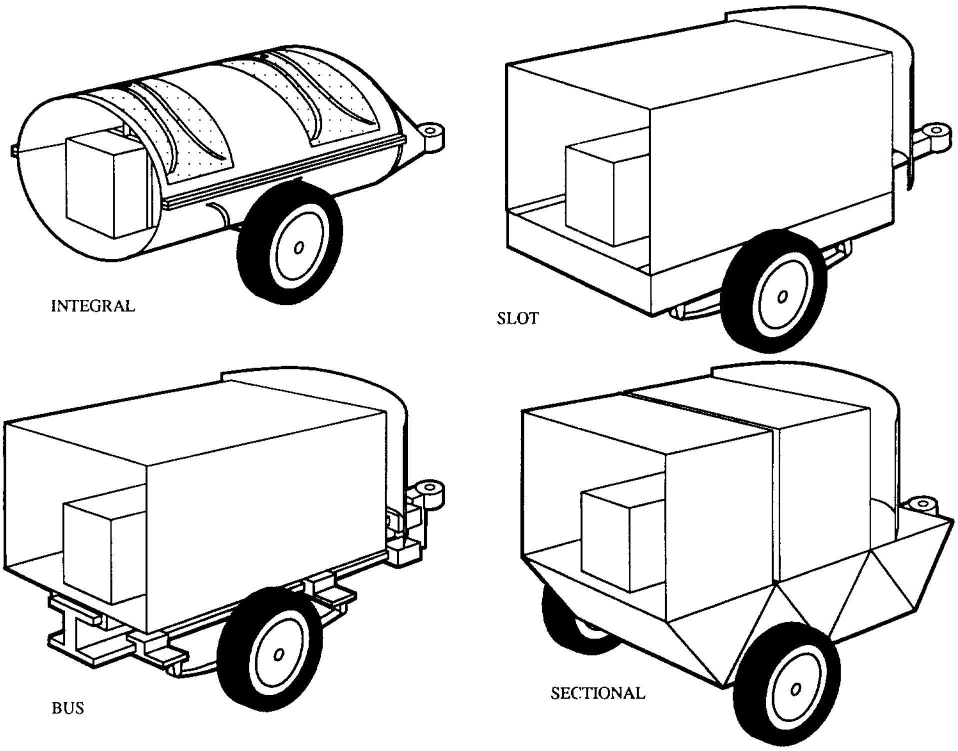

Fig. 5. Four trailer architectures.

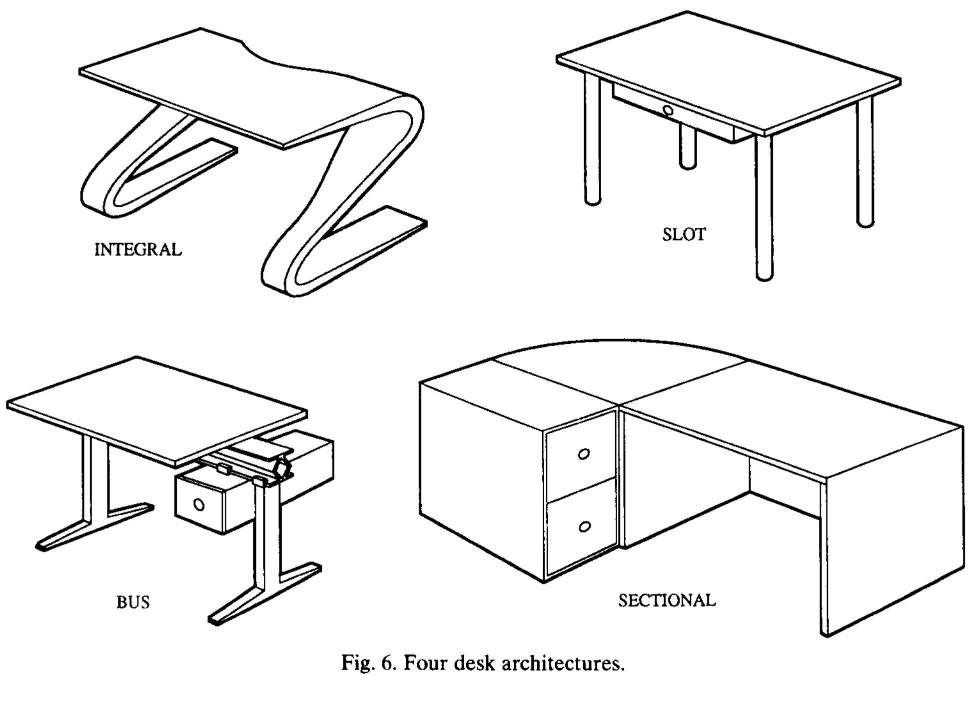

Fig. 6. Four desk architectures.

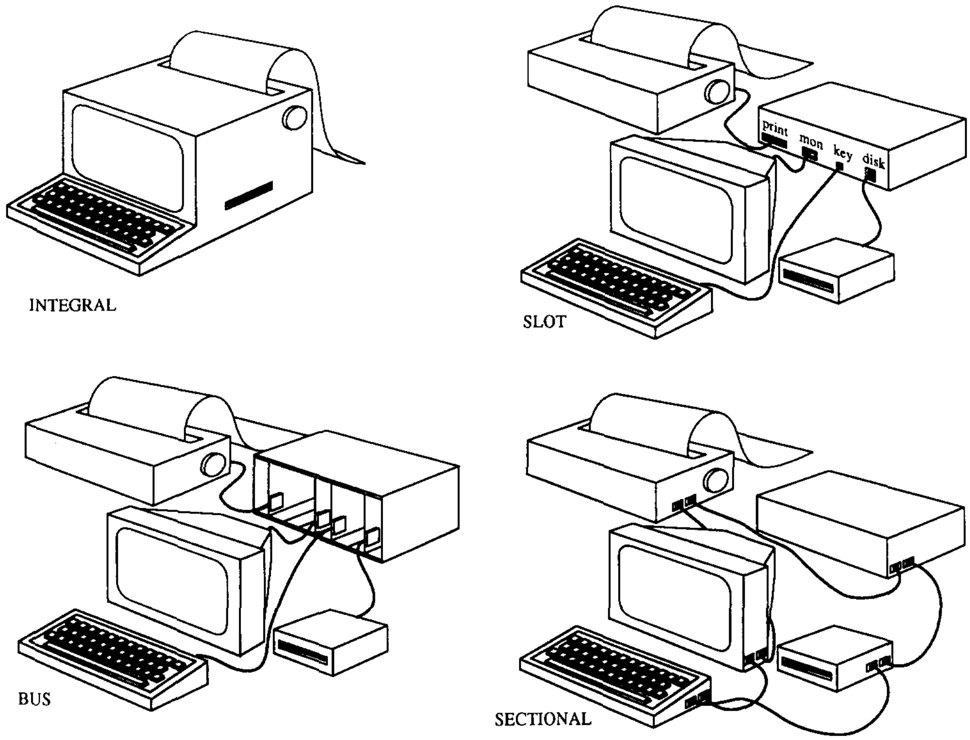

Fig. 7. Four personal computer architectures.

mal in all cases. The balance of the paper discusses the potential linkages between the architecture of the product and a set of issues of managerial importance. A recognition and understanding of these linkages is a prerequisite to the effective choice of an architecture for a particular product.

4. Product change

This section focuses on two types of product change: change to a particular artifact over its lifecycle (e.g. replacing a worn tyre) and change to a product line or model over successive generations (e.g. substituting the next generation suspension system in the whole product line). Section 5 and Section 6 treat two closely related concepts: product variety and component standardization.

4.1. Product architecture determines how the product can be changed

The minimum change that can be made to a product is a change to one component. The architecture of the product determines which functional elements of the product will be influenced by a change to a particular component, and which components must be changed to achieve a desired change to a functional element of the product. At one extreme, modular products allow each functional element of the product to be changed independently by changing only the corresponding component. At the other extreme, fully integral products require changes to every component to effect change in any single functional element. The architecture of a product is therefore closely linked to the ease with which a change to a product can be implemented. Here we consider how this linkage manifests itself in implementing change within the life of a particular artifact and in implementing change over several product generations.

4.2. Change within the life of a particular artifact

Products frequently undergo some change during their life. Some of the motives for this change are:

- Upgrade: As technological capabilities or user needs evolve, some products can accommodate this evolution through upgrades. Examples include changing the processor board in a computer printer or replacing a pump in a cooling system with a more powerful model. Some products, such as the Compaq Deskpro/M, have been promoted based on their ease of upgrade [26].
- Add-ons: Many products are sold by a manufacturer as a basic unit to which the user adds components, often produced by third parties, as needed. This type of change is common in the personal computer industry (e.g. the addition of third-party mass storage devices to a basic computer). See Langlois and Robertson [14] for a thorough description of several such cases.
- Adaptation: Some long-lived products may be used in several different-use environments, requiring adaptation. For example, machine tools may have to be converted from 220V to 110V power. Engines may have to be converted from a gasoline to a propane fuel supply.
- Wear: Physical features of a product may deteriorate with use, necessitating replacement of the worn components to extend the useful life of the product. For example, many razors allow dull blades to be replaced, tyres on vehicles can usually be replaced, most rotational bearings can be replaced, and many appliance motors can be replaced.
- Consumption: Some products consume materials that are typically replaceable. For example, copiers and printers frequently contain toner cartridges, cameras contain film cartridges, glue guns contain glue sticks, torches contain gas cartridges, and watches contain batteries.
- Flexibility in use: Some products can be configured by the user to exhibit different capabilities. For example, many 35 mm cameras can be used with different lens and flash options, some boats can be used with several awning options, and some fishing rods accommodate several rod-reel configurations.

In each of these cases, changes to the product are most easily accommodated through modular architectures. The modular architecture allows the required changes that are typically associated with the product's function to be localized to the minimum possible number of components.

Although consumption and wear is frequently accommodated through a modular design with replaceable parts, another popular strategy is to dramatically lower the cost of the entire product, often through an integral architecture, such that the entire product can be discarded or recycled. For example, disposable razors, cameras and cigarette lighters have all been commercially successful products, and disposable pens dominate the marketplace. Section 7 explains how integral architectures can allow for a lower cost product under certain conditions.

4.3. Change across generations of the product

When a new model of an existing product is introduced to the marketplace, the product almost always embodies some functional change relative to the previous product. (In relatively rare cases, the firm changes only the name of the product). The architecture of the product has profound implications for a firm's ability to implement this product change. For products with a modular architecture, desired changes to a functional element can be localized to one component. Products with integral architectures require changes to several components in order to implement changes to the product's function. This observation helps to explain industrial practice in the area of generational change.

For example, the Sony Walkman architecture allows the tape transport mechanism to be reused in many successive models, while the enclosure parts can be easily changed for each new model [24]. *Virtual design* is a term Sanderson [23] uses for this superposition of several product cycles involving changes to only a few components onto the longer life cycle of a technological platform.

This virtual design is enabled by the modular product architecture exhibited by the Walkman at the level of major subassemblies.

Sanchez and Sudharashan [22] describe a development strategy they call *real-time market research*. Under this scheme, the firm introduces a product, gauges the market response, then develops and launches an incrementally-improved product extremely quickly. A modular architecture is essential to being able to quickly change the product in this way. The benefits of a modular architecture for exploring a market and fine-tuning a product are also described in Langlois and Robertson [14].

Cusumano and Nobeoka [4], in summarizing several previous studies of the world automobile industry, identify *project scope* —the percentage of unique components a manufacturer designs from scratch in-house —as a key variable relating to product development performance. The architecture of the product, and the degree of modularity in particular, dictate how much project scope will be required to achieve a particular level of functional change.

In software engineering, change is notoriously difficult; Korson and Vaishnavi [13] find strong empirical evidence that modular software architectures facilitate program change. Change to a product is not always confined to activities by a single manufacturer. In some markets, such as home entertainment, users create *virtual products* by assembling collections of products provided by diverse manufacturers. Modularity at the level of the entire system, when combined with standard interfaces, allows for the virtual product to evolve and change through independent actions by individual manufacturers [14].

5. Product variety

I define product variety as the diversity of products that a production system provides to the marketplace. Product variety has emerged as an important element of manufacturing competitiveness. Based on survey responses from 255 managers, Pine [20,21] provides empirical evidence that both market turbulence and the need for product variety have increased substantially over the past decade and argues that variety will continue to increase in the future. Variety is also one of the elements of 'lean production', which has been identified as a successful approach to automobile manufacturing [36].

High variety can be produced by any system at some cost. For example, an auto manufacturer could create different fender shapes for each individual vehicle by creating different sets of stamping dies, each of which would be used only once. Such a system is technically feasible, but prohibitively expensive. The challenge is to create the desired product variety economically.

The ability of a firm to economically produce variety is frequently credited to manufacturing *flexibility*. (See Suarez et al [27] for a comprehensive review of the literature on flexibility.) When viewed at the level of the entire manufacturing system, this is a tautology —if a system is economically producing variety it is to some extent flexible. However, manufacturing flexibility is often equated with the flexibility of the process equipment in the plant (e.g. computer-numerical controlled milling machines), or with flexible assembly systems (e.g. programmable electronic chip insertion equipment). (See for example [12]) In this context, a flexible production process incurs small fixed costs for each output variant (e.g. low tooling costs) and small changeover costs between output variants (e.g. low set-up times). This notion of flexibility is consistent with Upton's definition [34]: "... the ability to change or adapt with little effort, time, or penalty". I argue that much of a manufacturing system's ability to create variety resides not with the flexibility of the equipment in the factory, but with the architecture of the product. This section shows how both the flexibility of the factory production process equipment and the product architecture interact to contribute to the ability to economically create product variety.

Variety is only meaningful to customers if the functionality of the product varies in some way [3].

[3] Functionality, in this context, is used broadly to mean any attribute of the product from which the user derives a benefit, and so would include, for example, styling or colour changes.

This variation may be in terms of the set of functional elements implemented by the product (Does the trailer protect the cargo from the environment at all?), or in terms of the specific performance characteristics of the product relative to a particular functional element (Is the environmental protection *normal* or *heavy duty*?). Consider the trailer example. Assume customers' needs can be neatly divided in the following ways. Some customers want to minimize air drag, some do not. Two types of vehicle connection and three alternatives for the type of environmental protection are desired. Three alternatives are also desired for both the structural load rating and for the ride quality of the suspension system [4]. Under these assumptions, if variety incurred no cost, the firm would offer 108 distinct trailers to the marketplace ($2 \times 2 \times 3 \times 3 \times 3 = 108$).

If the firm uses the modular product architecture shown in Fig. 2, all of the 108 different trailers can be created from a total of only 12 different types of components: a single type of fairing (which is either included with the trailer or not), two types of hitches, three types of boxes, three types of beds, three types of spring assemblies and one type of wheel assembly. Because each functional element maps to exactly one physical component, and because the interfaces are de-coupled, the variety can be created by forming 108 combinations from a set of 12 component building blocks. I am not the first to observe that variety can be created by combinations of building blocks. In fact, this combinatorial approach to variety is part of a five-step technique called (somewhat confusingly) Variety Reduction Program [29]. Nevins and Whitney [18] also give several examples of such combinatorial assembly of product variants. The modularity of the product allows the variety to be created at final assembly, the last stage of the production process. Some firms are even delaying a portion of the final assembly until the product has moved through the distribution system and is ready to be shipped to a customer. This strategy has been called *postponement* [15].

If the firm wishes to offer all 108 variants and uses the integral product architecture shown in Fig. 3, 73 different types of components will be required: 27 types of upper halves, 27 types of lower halves, 12 types of nose pieces, three types of cargo hanging straps, three types of spring slot covers and one type of wheel assembly. Because in many instances each component implements several functional elements, there must be as many types of each component as there are desired combinations of the functional elements it implements. For example, to provide all of the different desired combinations of the two vehicle connection types, the two types of drag reduction, and the three load ratings, 12 distinct types of nose pieces will be required because the nose piece contributes to all three of the functional elements associated with the options.

5.1. Variety and flexibility

At first glance, producing 108 varieties of the integral design appears to be far less economical than for the modular design. In fact, the flexibility of the production process equipment is an additional factor in determining the basic economics of producing variety. If the trailer components can only be economically produced in large lot sizes because of the large set up times required for the process equipment, or if each type of component required large tooling investments, then in fact the integral design would be very expensive to produce with high variety. High variety under these conditions would require some combination of large inventory costs, large set-up costs, or large tooling costs [5]. However, if the integral trailer components could be produced economically in small lots (e.g. set-up costs are low) and without tooling investments, then variety

[4] Assume for the purpose of the example that the type of suspension and the load rating are independent choices. In practice, these two functional elements may in fact be related.

[5] Inventory costs and set-up costs can be traded off against one another; inventory can be minimized by using small lot sizes, but this leads to high set-up costs.

could be offered economically for the integral design.

For example, consider the following production system for the integral trailer. The upper and lower halves are made by a computer controlled rolling machine followed by a computer controlled laser cutting machine. Plates of arbitrary thickness and material can be rolled to arbitrary diameters (within certain limits), and slots for the springs can be cut along arbitrary trajectories; all with small set-up times, no tooling investment, and rapid processing times. The nose piece is created by laser cutting, computer-controlled rolling and automated welding. The six components are then assembled manually. Because of the flexibility of the upper half, lower half and nose piece production processes, the required component types can be produced as they are needed in arbitrary combinations, and then assembled into the required trailer types. Such process flexibility allows economical high-variety production of a product with an integral architecture.

Flexible production process hardware can also have an impact on the production of the modular design. Using inflexible processes requiring expensive tooling and large lot sizes, the 12 different components required to assemble the 108 different product variants would be held in inventory ready for final assembly. Alternatively, the components for the modular design could be produced with flexible production equipment, eliminating the need for the inventories and tooling expense.

With a modular product architecture, product variety can be achieved with or without flexible component production equipment. In relative terms, in order to economically produce high variety with an integral architecture, the component production equipment must be flexible.

This argument assumes in all cases that the final assembly process itself is somewhat flexible, i.e. different combinations of components can be easily assembled to create the final product variety. This assumption is usually valid for products assembled manually, but some assembly systems, particularly high-volume automated assembly equipment, violate this assumption. For these systems, the flexibility of the final assembly process is also a key driver of the ability of the firm to offer product variety.

5.2. *Infinite variety*

Many flexible production processes can be programmed to produce an infinite variety of components. For example, a computer-controlled laser cutting system can cut along an arbitrarily specified trajectory. This flexibility allows systems incorporating these processes to create products that can be infinitely varied with respect to one or more properties. This ability to continuously vary the properties of components by a flexible process provides a subtle distinction between the variety that can be created by assembling products from a finite set of component alternatives, and the variety that can be created by flexible component production processes. Assembly from finite component choices is fundamentally a 'set operation', in that it allows sets to be formed from discrete alternatives. Continuously variable process equipment can implement arbitrary mathematical relationships among component characteristics. For example, the laser cutting machine could be programmed to cut along a curve parameterized as a function of a set of other

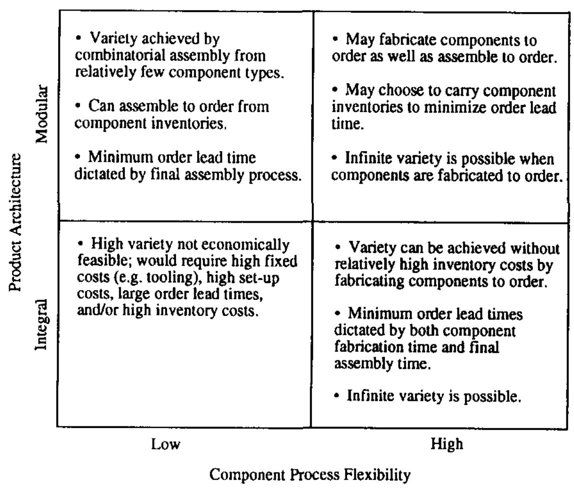

Fig. 8. Product architecture *and* component process flexibility dictate the economics of producing variety.

characteristics, such as expected climate of the use environment, the types of loads the trailer will carry, and the road quality in the customer's geographical region. Note that the ability to arbitrarily vary component characteristics can be achieved for both integral and modular architectures if components are fabricated with programmable processes.

A summary of the effect of product architecture and component process flexibility on the resulting performance characteristics of the production system is shown in Fig. 8.

6. Component standardization

Component standardization is the use of the same component in multiple products and is closely linked to product variety. Common standardized components include tyres, batteries, bearings, motors, light bulbs, resistors and fasteners. Component standardization occurs both within a single firm (e.g. Quad-4 engines at General Motors) and across multiple firms (e.g. Timken roller bearings at Ford, General Motors, and Chrysler). I call the first case *internal* standardization, and the second case *external* standardization. For internal standardization, components may be designed and manufactured within the firm or provided by suppliers. For external standardization, components are typically designed and manufactured by suppliers.

6.1. A modular architecture makes standardization possible

Standardization can arise only when: (a) a component implements commonly useful functions; and (b) the interface to the component is identical across more than one different product. Otherwise, a component would either not be useful in more than one application or would not physically fit in more than one application.

A modular architecture increases the likelihood that a component will be commonly useful. When the mapping from functional elements to components is one-to-one, each component implements one and only one function. Such components are therefore useful in any other product applications where their associated functions occur. Components of a product exhibiting an integral architecture would only be potentially useful in other products containing the exact combination of functional elements, or parts of functional elements, implemented by the component.

A modular architecture also enables component interfaces to be identical across several products. Interfaces in modular architectures are decoupled, i.e. a particular component will not have to change when surrounding components are changed. Therefore, different sets of surrounding components, such as might occur in different applications, do not require different component interfaces. When interfaces are decoupled, an interface standard can be adopted and the same component can be used in a variety of settings.

6.2. What are the implications of standardization?

Component standardization, whether external or internal, has implications for the manufacturing firm in the areas of cost, product performance and product development.

Under most circumstances a standard component is less expensive than a component designed and built for use in only one product. This lower cost is possible primarily because the standard component will be produced in higher volume, allowing greater economies of scale and more learning. Higher component volume may also attract several competitors who exert price pressure on one another. When external standardization occurs, this cost advantage can be viewed in economic terms as a network externality [5,6]. However, there are some circumstances under which the use of a standard component may incur higher unit costs than the use of a special component. Sometimes in an effort to standardize, firms will use a component with excess capability for a particular application. For example, a standard enclosure may be slightly larger than necessary in a particular application, or a standard power supply may provide slightly more power than is strictly necessary in a particular application. In these cases, firms may choose to adopt the standard components even if their unit cost is higher than that of a component more closely matched to the application. This standardization may be justifiable because of the economic savings from reduced complexity in, for example, purchasing, inventory management, quality control or field service.

Standard components, in general, exhibit higher performance (for a given cost) than unique designs. This performance advantage arises from the learning and experience the component supplier is able to accumulate. However, standardization may act as an inertial force preventing firms from adopting a better component technology because of compatibility issues in the installed base of products [5,6].

The use of standard components can lower the complexity, cost and lead time of product development. An existing standard component represents a known entity and therefore can reduce the number of uncertain issues the development team must cope with. An existing standard component also requires no development resources and so can lower both the cost and, if the component development would have been on the project critical path, the lead time of a project.

7. Product performance

I define product performance as how well the product implements its functional elements. Typical product performance characteristics are speed, efficiency, life and noise. Product performance, as defined here, excludes *economic* performance, except to the extent that it arises from the product's technical performance, because economic performance is also highly dependent on the firm's production, service, sales and marketing activities.

Some performance characteristics arise only from the physical properties of a local region of the product. For example, the intensity of light from the tail of the trailer is a performance characteristic that arises only from the physical properties of those components implementing the aft illumination function. I call such characteristics *local performance characteristics*.

In contrast, many performance characteristics of a product arise inevitably from the physical properties of most, if not all, of the components of the product. These *global performance characteristics* are tied to the product's size, shape, mass and material properties. For example, vehicle fuel efficiency arises from, in addition to the trailer's aerodynamic profile, the trailer's mass. Mass is inevitably determined by every atom in the product. Other typical global performance characteristics include electromagnetic emissions, balance, aesthetics, power consumption, noise and vibration.

Local performance characteristics can be optimized through a modular architecture, but global performance characteristics can only be optimized through an integral architecture.

7.1. Local performance characteristics and modular architectures

Modular architectures allow for optimization of local performance characteristics for practical, more than for theoretical, reasons. First, as discussed in Section 6, a modular architecture may allow the use of a standard component. The use of a standard component allows the firm to exploit the performance refinements the supplier of this component has been able to make over the entire history of the component's use. Second, even when a standard component is not available and a component must be developed from scratch, a modular architecture allows the component to be designed, tested and refined in a focused way without disruptions and distractions arising from the need to address either interface coupling or other functional elements. All other things being equal, these benefits in design, testing and refinement lead to higher component performance. This explains why a trailer manufacturer trying to optimize light intensity or tyre life would likely adopt an architecture allowing the use of modular lamp and tyre components.

Note that what may be considered a component of one product is itself a product or system for the supplier of the component (whether the supplier is internal or external). As a result, the component itself may be designed with a highly integral architecture, but then may be used in a highly modular way as part of a larger product or system. For example, tyres exhibit a highly integral architecture, but may be used as a component in a trailer with a highly modular architecture.

7.2. *Global performance characteristics and integral architectures*

All physical products occupy space, exhibit some shape, and are composed of materials with mass and other physical properties. I illustrate the role architecture plays in global performance with the specific case of optimizing performance by minimizing the size and mass of a product; similar arguments can be made about other physical properties, such as natural frequency of vibration or electromagnetic radiation.

For most products, several key performance characteristics are closely related to the size and shape of the product or to its mass. For example, acceleration relates to mass, aerodynamic drag relates to size and shape, and, for our example, vehicle fuel efficiency relates to size and shape as well as to mass. In most cases, increasing global performance characteristics involves decreasing size and mass. (In relatively rare cases, increasing global performance involves increasing size and mass; improving the holding power of a boat anchor or increasing the passenger comfort of an automobile may be such cases.)

Two design strategies are frequently employed to minimize mass or size: *function sharing* and *geometric nesting*. Function sharing is a design strategy in which redundant physical properties of components are eliminated through the mapping of more than one functional element to a single component [33]. For example, a conventional motorcycle contains a steel tubular frame distinct from the engine and transmission. In contrast, several high-performance motorcycles contain no distinct frame. Rather, the cast aluminum transmission and motor casing acts as the structure for the motorcycle. (See, for example, the photograph of the BMW R1100RS in Ulrich and Eppinger [30a]) The motorcycle designers adopted function sharing as a means of exploiting the fact that the transmission and motor case had incidental structural properties which were redundant to the structural properties of the conventional frame. Through function sharing the designers minimize the mass of the frame/motor/transmission system. In exploiting the secondary structural properties of the motor and transmission case, the designers mapped more than one functional element to a single component and therefore created an integral architecture.

Geometric nesting is a design strategy for efficient use of space and material and involves the interleaving and arrangement of components such that they occupy the minimum volume possible, or, in some cases, such that they occupy a volume with a particular desired shape. For example, the wheel, suspension, fender and brake system of a modern automobile are arranged in a way that barely allows clearance for wheel travel; they are tightly nested. An unfortunate consequence of nesting is the coupling of the interfaces between components, the other hallmark of an integral architecture. For example, in an automobile the brake system cooling is tightly coupled to the shape of the wheel well, the wheel covers and the fenders. A slight change to the shape of the wheel cover can require substantial changes to the brake disc design. Similarly, the road and wind noise from the wheels is coupled in a complex way to the shape of the wheel well and fender. Thus, a desire for increased global performance in the area of drag and aesthetics leads to a design strategy of geometric nesting. This design strategy causes components to be coupled, thereby sacrificing the modularity of the product architecture.

Minimizing size and mass is also part of a strategy for minimizing unit production costs for high-volume products, because as production volumes increase materials costs become more and more significant. This explains why integral architectures are sometimes employed to achieve very low unit costs, such as are required for disposable products like ball-point pens, razors and single-use cameras.

The examples in this section illustrate extreme conditions. Most products or systems will embody hybrid modular-integral architectures. For example, although the high-performance motorcycle may exhibit little modularity in the architecture of the engine, transmission and frame, the architecture of the ignition system may be quite modular (e.g. spark plug, wiring, coil, etc.). The designers of the motorcycle have avoided modularity only where the global performance penalties are most severe.

This view of how architecture relates to performance is another perspective on the notion of *product integrity* articulated by Clark and Fujimoto [3]. Product integrity can be viewed as the result of optimizing global performance characteristics. This optimization requires an integral architecture for some regions of the product, which in turn requires specific managerial approaches and techniques during new product development.

8. Product development management

At a basic level the product development process can be viewed as consisting of four phases: concept development, system-level design, detailed design, and product testing and refinement. The activities of the concept development phase include: the selection of the technological working principles of a product; the choice of functional elements, features and performance targets in order to best meet customer needs; and a choice of architectural approach. The system-level design phase includes the development of the product architecture and the assignment of component development tasks to the extended product development team. The detailed design phase is primarily concerned with component design, testing and production process planning. The product testing and refinement phase involves assembling and testing prototypes and implementing any required changes to the component designs.

The architecture of the product has implications for the effectiveness of approaches to the three development phases following concept development. The following sections discuss these three phases and Fig. 9 summarizes the differences in effective approaches for modular and integral architectures.

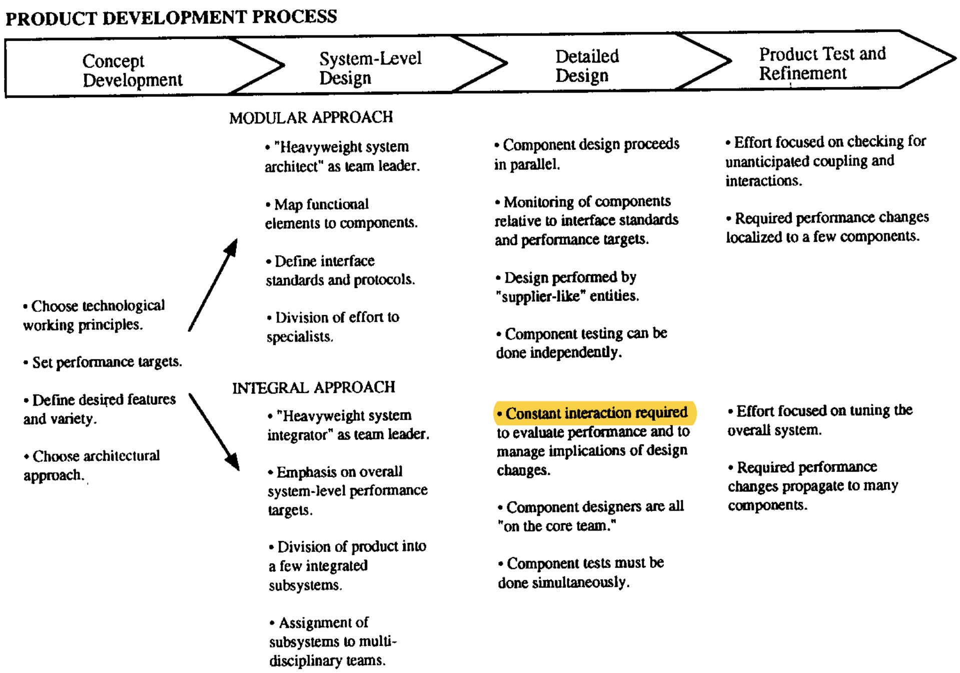

Fig. 9. Differences in product development management according to architectural approach.

8.1. System-level design

A modular architecture requires relatively more emphasis on this phase of development than does an integral architecture. For the modular architecture the focus of system-level design and planning is to carefully define component interfaces, specifying the associated standards and protocols. Performance targets and acceptance criteria are set for each component, corresponding to the particular functional element implemented by the component. Component design is frequently assigned to specialists, either internal or external to the firm. The development team leader can be viewed as a 'heavyweight system architect' [6].

For the integral architecture, system-level design absorbs relatively less effort. The focus is on establishing clear targets for the performance of the overall system and on dividing the system into a relatively small number of integrated subsystems. These subsystems are frequently assigned to multi-disciplinary teams who will share the responsibility for designing the components that make up the subsystem. The leader of these teams can be viewed as a 'heavyweight system integrator'.

8.2. Detailed design

For the modular architecture, detailed design of each component can proceed almost independently and in parallel. Management of the detailed design process consists of monitoring the progress of each individual component design activity relative to the component performance targets and interface specifications. The component design teams are 'supplier-like' in that interaction is structured and relatively infrequent. Testing of each component can be performed independently and clear objectives define completion of each component design activity.

For the integral architecture, component designers all form a 'core team' and interact continually in order to analyze performance of the subsystem to which their component belongs and to manage changes required because of component interface coupling. Whether the components meet their performance targets depends on their interaction and not on whether they meet some pre-specified criteria. Testing of components cannot be completed in isolation; subsystems of components must be assembled and tested as a whole.

8.3. Product test and refinement

For the modular product, product testing and refinement is a checking activity. The tests are intended to detect unanticipated interactions among the components. These interactions are viewed as 'bugs' and their resolution is usually localized to changes to one or two components.

For the integral product, product testing and refinement is a tuning activity. If the product performance must be altered in some way, changes are likely to be required to many components. Relatively more time will be spent in this phase than for the modular product.

8.4. Organizational implications

There are at least three organizational issues tied to a choice of architectural approach: skills and capabilities, management complexity, and the ability to innovate.

Highly modular designs allow firms to divide their development and production organizations into specialized groups with a narrow focus. This organizational structure may also extend to the supplier network of the firm. If the function of a component can be precisely specified and the interface between the component and the rest of the product is fully characterized, then the design and production of that component can be assigned to a separate entity. Such specialization may facilitate the development of deep expertise relative to a particular functional element and its associated component.

Required project management skills are different for different architectures. Modular architectures may require better systems engineering and planning skills, while integral architectures may require better coordination and integration skills.

[6] This term is meant to complement the notion of a "heavyweight project manager" articulated by Wheelwright and Clark [35a].

Table 1
Summary of key ideas

	Integral	Modular-Slot	Modular-Bus	Modular–Sectional
Definition	• Complex mapping from functional elements to components. • And/or the component interfaces are coupled.	• One-to one mapping between functional elements and components. • Interfaces between components are not coupled.		
		• Component interfaces are all different.	• Component interfaces are all the same.	
			• A single component (the bus) links the other components.	
Examples	• Automobile unit body.	• Truck body and frame.		
	• Neon sign/lighting.	• Table lamp with bulb and shade.	• Track lighting.	
			• Shelves with brackets and rails.	• Stackable shelving units.
	• "Boom Box" (some internal components are modular-slot).	• Consumer component stereo.	• Professional audio equipment in 19 inch rack.	
	• Cargo ship (hull in particular).	• Tractor-trailer.		• Freight train.
Product Change	• Any change in functionality requires a change to several components	• Functional changes can be made to a product in the field. • Manufacturers can change the function of subsequent model generations by changing a single component.		
Product variety	• Variety not feasible without flexible component production processes.	• Products can be assembled in a combinatorial fashion from a relatively small set of component building blocks to create variety. • Variety possible even without flexible component production processes.		
		• Variety confined to the choices of components within a pre-defined overall product structure.		• Variety in overall structure of the product possible (e.g. Lego blocks, piping).
Component Standardization		• Components can be standardized across a product line. • Firms can use standard components provided by suppliers. • Interfaces may adhere to an industry standard.		
Product Performance	• May exhibit higher performance for global performance characteristics like drag, noise, and aesthetics.	• May facilitate local performance. • Decoupling interfaces may require additional mass and space. • One-to-one mapping of functional elements to components prevents. *function sharing*–the simultaneous implementation of more than one functional element by a single component–potentially resulting in physical redundancy		
			• Standardized interfaces may result in additional redundancy and physical "overhead".	
Product Development Managment	• Requires tight coordination of design tasks.	• Design tasks can be cleanly separated, thus allowing the tasks to be completed in parallel. • Specialization and division of labor possible. • Architectural innovation may be difficult. • Requires the top-down creation of a global product architecture.		

Firms with a long history of a particular architectural approach are likely to have developed the associated skills and capabilities.

A modular architecture enables a bureaucratic approach to organizing and managing development. This approach, for relatively well understood technologies, allows the complexity of the product development process to be dramatically reduced and may allow for better exploitation of supplier capabilities. Lovejoy articulates the highly non-linear theoretical reduction in complexity engendered by decomposing the design problem into de-coupled subproblems [16]. Von Hippel [35] argues that problem decomposition, and by implication product architecture, is important in managing development projects. Clark [2] provides evidence that automobile manufacturers with the shortest product development times adopt a 'black box' approach to component development, in which the basic function of a component as well as its interfaces are specified, but the details of the design are not. For some domains the benefits of reduced complexity and enhanced supplier involvement may drive the choice of the architecture for at least parts of the product; software development is one such domain. In most cases the system-level performance penalties of a modular architecture are dwarfed by the benefits of a reduction in project management complexity.

A potential negative implication of a modular product architecture is the risk of creating organizational barriers to architectural innovation. These barriers appear to be unfortunate side effects of focus and specialization. This problem has been identified by Henderson and Clark [10] in the photolithography industry and may in fact be of concern in many other industries as well.

9. Closing remarks

The overarching message of this paper is that manufacturing firm performance is linked to the architecture of the product. Product architecture consists of: (1) the arrangement of functional elements, or the function structure; (2) the mapping from functional elements to physical components; and (3) the specification of the interfaces between interacting components. Table 1 summarizes the key ideas in the paper. This closing section discusses how to establish a product architecture, identifies three research directions, and draws a few conclusions.

9.1. How to establish a product architecture

Dozens of issues are linked to the architecture of the product. The net effect is a complex set of relations among many areas of concern. While there are currently no deterministic approaches to choosing an optimal product architecture, the process can be guided. In most cases the choice will not be between a completely modular or completely integral architecture, but rather will be focused on which functional elements should be treated in a modular way and which should be treated in an integral way. Listed here are questions the product development team and firm management can ask in order to raise the important issues and to guide the development of an appropriate architecture. These questions are best posed during the concept development phase of the product development process. These questions also serve as a summary of the linkages between product architecture and the areas of managerial concern described in Sections 4 through 8.

9.1.1. Product change

- Which functional elements are likely to require upgrade?
- Are third-party add-ons desirable?
- Which functional elements may have to be adapted to new use environments over the life of the product?
- Which functional elements will involve wear or consumption?
- Where will flexibility in configuration be useful to the user?
- Which functional elements can remain identical for future models of the product?
- Which functional elements must change rapidly to respond to market or technological dynamics?

9.1.2. Product variety

- Which variants of the product are desirable to best match variation in customer preferences?

- What level of flexibility of component process is available or easily obtained?
- How much advantage does minimizing order lead time for custom products provide?

9.1.3. Component standardization

- Are existing components available internally or externally for any of the functional elements of the product?
- What are the cost implications of sharing a component with another product?
- Where can adopting a standard component reduce development time or complexity of project management?

9.1.4. Product performance

- Which local performance characteristics are of great value to customers and can therefore be optimized through a modular architecture?
- Which global performance characteristics are of great value to customers and can therefore be optimized through an integral architecture?

9.1.5. Product development management

- How much focus and specialization is present in the organization and in the supplier network?
- Is the product inherently large and complex?
- Is the development team geographically dispersed?
- Are barriers to architectural innovation developing in the organization because of specialization?
- Has the organization demonstrated an ability to change in structure and style?

9.2. Research directions

The research described in this paper is conceptual and foundational. My approach has been to synthesize fragments from several different disciplines, including software engineering, design theory, operations management and product development management. I have tried to create a coherent definition of product architecture and to use logical arguments and examples to illuminate the linkages between product architecture and important issues facing manufacturing firms. I hope to have motivated a set of problems and issues, but much analytical and empirical work remains. Three research directions seem particularly interesting and important.

First, the need to make decisions involving trade-offs motivates the development of decision support models. A single model of most of the trade-offs associated with the choice of a product architecture is unlikely, and even if it were developed would probably be too complex to be useful. However, focused problems can probably be usefully isolated, analyzed, and modeled. For example, a model integrating marketing science ideas (such as those in [9]) and production cost models could be used to evaluate the optimal variety that should be produced for each of two product architectures: integral and modular. The integral and modular architectures would each have their own cost structure and would likely lead to different levels of optimal product variety. Such a model could be used to coordinate systems engineering decisions involving product architecture, with market segment information and production cost information. Similar models could be built to support decisions involving component standardization, investments in production process flexibility and order lead time.

Second, I believe much insight would be gained by conducting an empirical study of the elements of difference in product architectures among the products manufactured by different firms. Such a study might lead to an identification of factors that dominate the choice of a product architecture. The results might also lead to an identification of multiple, equally effective, strategies involving different combinations of product architectures, organizational structures and production systems. I have used a methodology I call *product archaeology*, meaning the study of the physical artifact itself, to better understand design-for-manufacturing decision making [31]. This approach could also be applied to understanding the differences in product architectures among products from different manufacturers.

Finally, there is some evidence that the organization of the firm and the architecture of the product are interrelated. This linkage seems worthy of further research. Several specific questions could be addressed. Does the existence of a strong component supplier industry drive firms to organize in a particular way and to adopt a particular architecture? Do vertically integrated firms adopt more or less modular designs than firms working with outside suppliers? Does firm size or geographic location relate to the architecture of the product? Are firms able to change the architecture of their products without changing their organizational structure? If so, which organizational structures allow the most flexibility in product architecture.

9.3. Conclusions

While the concept of an explicit product architecture is prevalent in large electronic systems design and in software engineering, to my knowledge relatively few manufacturers of mechanical and electromechanical products explicitly consider the architecture of the product and its impact on the overall manufacturing system. Hopefully, the ideas in this paper will be useful, first, by raising the awareness of the far-reaching implications of the architecture of the product, and second, by creating a vocabulary for discussing and addressing the decisions and issues that are linked to product architecture.

In addition to providing a conceptual framework, I hope that by enumerating and discussing specific trade-offs the paper contributes directly to the decisions made during the concept development and systems engineering phases of product development. These decisions include: Which variants of the product will be offered in the marketplace? How will the product be decomposed into components and subsystems? How will development tasks be allocated to internal teams and suppliers? What combination of process flexibility and modular product architecture will be used to achieve the desired product variety?

In the 1980s much attention was focused on the relationship between product design and manufacturing. While in many cases this attention led to improvements in production costs, it was focused on designing products to be easy to assemble and on reducing the cost of individual piece parts. The linkages between the product and the performance of the manufacturing firm are in fact much more extensive and include the relationship between the architecture of the product and the way the product will be changed, the variety offered in the marketplace, component standardization, the performance of the product and the management of product development.

Acknowledgments

This research was supported by the National Science Foundation under contract number DDM-8914181 and by the MIT Leaders for Manufacturing Program, a partnership among MIT and 12 major corporations. Many of the ideas in the paper were developed during a collaborative research effort with the Square D Company. I am grateful to William Fonte, David Ellison, Steven Eppinger, George Favaloro, Frank Gillett, Stephen Graves, Rebecca Henderson, William Lovejoy, Wanda Orlikowski, B. Joseph Pine, Kamalini Ramdas-Jain, Henry Stoll, Bala Subramaniam, Mohan Tatikonda, Marcie Tyre and Daniel Whitney for their comments on earlier versions. The Research Policy reviewers also provided valuable recommendations for revision. Some of the ideas in this paper grew out of a previous paper with Karen Tung, Fundamentals of Product Modularity, Proceedings of the 1991 ASME Winter Annual Meeting Symposium on Issues in Design/Manufacturing Integration (ASME DE-Vol. 39).

References

[1] Christopher Alexander, *Notes on the Synthesis of Form*, (Harvard University Press, Cambridge, 1964).

[2] Kim B. Clark, Project Scope and Project Performance: Effect of Parts Strategy and Supplier Involvement on Product Development, *Management Science* 35(10) (1989) 1247–1263.

[3] Kim B. Clark and Takahiro Fujimoto, The Power of Product Integrity, *Harvard Business Review* 68(6) (1990).

[4] Michael A. Cusumano and Kentaro Nobeoka, Strategy, Structure and Performance in Product Development:

Observations from the Auto Industry, *Research Policy* 21 (1992) 265–293.

[5] Joseph Farrell and Garth Saloner, Standardization, Compatibility, and Innovation, *Rand Journal of Economics* 16(1) (1985) 70–83.

[6] Joseph Farrell and Garth Saloner, Installed Base and Compatibility: Innovation, Product Preannouncements, and Predation, *The American Economic Review* 76(5) (1986) 940–955.

[7] Susan Finger and John R. Dixon, A Review of Research in Mechanical Engineering Design, *Research in Engineering Design* 1(1,2) (1989).

[8] Theodore Fowler, *Value Analysis in Design* (Van Nostrand Reinhold, New York, 1990).

[9] Paul E. Green and Abba M. Krieger, Models and Heuristics for Product Line Selection, *Marketing Science* 4(1) (1985).

[10] Rebecca M. Henderson and Kim B. Clark, Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms, *Administrative Science Quarterly* 35 (1990) 9–30.

[11] Vladimir Hubka and W. Ernst Eder, *Theory of Technical Systems* (Springer-Verlag, New York, 1988).

[12] R. Jaikumar, Postindustrial Manufacturing, *Harvard Business Review* 64(6) (1986) 69–76.

[13] Timothy D. Korson and Vijay K. Vaishnavi, An Empirical Study of the Effects of Modularity on Program Modifiability, Technical Report CIS-86-001, Georgia State University, Department of Computer Information Systems, 1986.

[14] Richard N. Langlois and Paul L. Robertson, Networks and Innovation in a Modular System: Lessons from the Microcomputer and Stereo Components Industries, *Research Policy* 21 (1992).

[15] Hau Lee and Corey Billington, Designing Products and Processes for Postponement (paper presented at the Conference on Design Management, Anderson Graduate School of Management, UCLA, September 17–18, 1992).

[16] William S. Lovejoy, Rationalizing the Design Process (paper presented at the Conference on Design Management, Anderson Graduate School of Management, UCLA, September 17–18, 1992).

[17] Carver Mead and Lynn Conway, *Introduction to VLSI Systems* (Addison-Wesley, Reading, MA, 1980).

[18] James L. Nevins and Daniel E. Whitney, *Concurrent Design of Products and Processes* (McGraw-Hill, 1989).

[19] Gerhard Pahl and Wolfgang Beitz,, in: Ken Wallace (Editor), *Engineering Design* (The Design Council, London, 1984).

[20] B. Joseph Pine II, Paradigm Shift: From Mass Production to Mass Customization, S.M. Thesis, MIT Sloan School of Management (1991).

[21] B. Joseph Pine II, *Mass Customization: The New Frontier in Business Competition* (Harvard Business School Press, 1992).

[22] Ron Sanchez and D. Sudharshan, Real-Time Market Research: Learning-by-Doing in the Development of New Products, *Proceedings of the International Conference on Product Development Management* Vol. II, Institute for Advanced Studies in Management, Brussels (1992).

[23] Susan Sanderson, Cost Models for Evaluating Virtual Design Strategies in Multicycle Product Families, *Journal of Engineering and Technology Management* 8 (1991) 339–358.

[24] Susan Sanderson and Mustafa Uzumeri, *Managing Product Families: The Case of the Sony Walkman*, Working Paper, Rensselaer Polytechnic Institute (1992).

[25] Stephen R. Schach, *Software Engineering* (Richard Irwin, Boston, 1990).

[26] Sean Silverthorne, Upgradable PCs Spreading, But Benefits are Debated, *Investor's Daily*, September 17 (1991).

[27] Fernando F. Suarez, Michael A. Cusumano and Charles H. Fine, Flexibility and Performance: A Literature Critique and Strategic Framework (Massachusetts Institute of Technology Sloan School of Management Working Paper 3298-91-BPS, 1991).

[28] Nam P. Suh, *The Principles of Design* (Oxford University Press, 1990).

[29] Toshio Suzue and Akira Kohdate, *Variety Reduction Program* (Productivity Press, Cambridge, MA, 1990).

[30] David G. Ullman, *The Mechanical Design Process* (McGraw-Hill, New York, 1992).

[30a] Karl T. Ulrich and Steven D. Eppinger, *Product Design and Development*, McGraw-Hill, New York, 1995.

[31] Karl T. Ulrich and Scott Pearson, Does Product Design Really Determine 80% of Manufacturing Cost? (Massachusetts Institute of Technology Sloan School of Management working paper, 3601-93-MSA 1993).

[32] Karl T. Ulrich and Warren P. Seering, Synthesis of Schematic Descriptions in Mechanical Design, *Research in Engineering Design*, 1 (1989) 3–18.

[33] Karl T. Ulrich and Warren P. Seering, Function Sharing in Mechanical Design, *Design Studies* 11 (1990) 223–234.

[34] David M. Upton, The Management of Manufacturing Flexibility: Work in Progress, Harvard Business School Working Paper 92-058, September 1991.

[35] Eric von Hippel, Task Partitioning: An Innovation Process Variable, *Research Policy* 19 (1990) 407–418.

[35a] Steven. C. Wheelwright and Kim B. Clark, *Revolutionizing Product Development: Quantum Leaps in Speed, Efficiency and Quality*, The Free Press, New York, 1992.

[36] J.P. Womack, D.T. Jones and D. Roos, *The Machine that Changed the World* (Rawson/MacMillan, New York, 1990).