

*The International Center for Research on the
Management of Technology*

**Is The Make-Buy Decision Process A Core
Competence?**

**Charles H. Fine
Daniel E. Whitney**

January 1996

WP # 140-96

Sloan WP # 3875

© 1996 Massachusetts Institute of Technology

Sloan School of Management
Massachusetts Institute of Technology
38 Memorial Drive, E56-390
Cambridge, MA 02139

Acknowledgments

This paper reflects many discussions with numerous of our MIT students, including Geoffrey Parker, Sharon Novak, and Nitin Joglekar. This paper also reflects numerous domestic and international visits and research trips funded by MIT programs (Leaders for Manufacturing, International Motor Vehicle Program, Industrial Performance Center, Japan Program, International Center for Research on the Management of Technology), as well as by ONR. We are particularly indebted to contacts with companies and interns sponsored by MIT's Leaders for Manufacturing program and the International Motor Vehicle Program.

"We are totally dependent on our suppliers."¹

1. Introduction and Overview

Many of today's products are so complex that no single company has all the necessary knowledge about either the product or the required processes to completely design and manufacture them in-house. As a result most companies are dependent on others for crucial elements of their corporate well-being. Typically, however, companies have some choice as to whom they become dependent upon and for what sorts of skills and competences. That is, although few companies can "do it all," most have significant influence over the strategic choice of corporate identity and what businesses to be in. What is the range of choices they face? How are different companies making those choices? Can we make sense of the variety of decisions we can observe now in different industries or different parts of the world? What are the skills that companies must retain?

In this paper we address the challenge of making these choices rationally. We give examples in which similar companies, facing similar choices, select make/buy patterns in very different ways, resulting in very different patterns of interdependencies along companies' supply chains. These choices are not restricted to skills related to the product, but include choices related to key design and manufacturing issues. To make sense of these differences, we propose a framework that ties together the following engineering and management concepts into one coherent view:

- core competencies
- the product development process
- systems engineering
- product architecture and modularity
- supply chain design

and seek to show that the technical and managerial skills required for each of these are highly overlapping if not the same.

¹Declared with pride by Tracy O'Rourke, CEO, Varian Associates, in his Keynote Address to the 4th Agility Forum, March 7, 1995

An important purpose of this paper is to broaden the discussion of core competence, considering *product* competencies, design and *manufacturing process* competencies, *business process* competencies, and the *dynamic instabilities* of the set of capabilities that are perceived at any one time to be core. We also discuss widely observed differences in how various companies prioritize product and process competencies and we try to make sense of these observations. We distinguish two categories of dependency, each with advantages and risks that can be used to define key distinctions in firms' make/buy strategies and core competence investments. In addition, we develop a model of the potentially unstable dynamics of industry integration. Our ultimate goal is to convince readers that the main skills companies should retain transcend those directly involving product or process, and are in fact the skills that support the very process of choosing which skills to retain.

The state of our understanding of this issue does not permit pat answers. The options open to companies are not wholly comfortable, and it is not obvious what is the right thing to do in each case. The underlying technology of the product or its supporting design and manufacturing processes may dictate the choice, leaving the company without an alternative. Sometimes the available choices do not result in a stable situation but rather leave the company, indeed sometimes an entire industrial sector, in a constant state of flux, with leadership and financial viability cycling from firm to firm. We provide a model suggesting why these situations exist.

The views expressed in this paper are somewhat speculative, representing an attempt to put structure onto a wide variety of anecdotal observations and tie together strands of prior research. An important theoretical forerunner of the ideas presented here is that of the distinction between innovation in product architecture and that in product components [Ulrich], as well as the different kinds of corporate organization required in each case. [Henderson and Clark] This paper builds on these ideas but is more specific about certain aspects of the problem. In particular, we argue that the ability to manage product design and manufacture from the platform of the architecture is an important skill, and that this skill permits a firm to encompass the components and their manufacture in a consistent way.

The paper is structured as follows: Section 2 develops the concept of *manufacturing infrastructure* as an foundational business element and describes stark differences in the make/buy policies and strategic treatment of this element by seemingly similarly-situated firms. Section 3 reflects on the observations of section 2. Section 4 describes the product realization process in which the make/buy decision is often embedded. In Section 5, we discuss the concepts of product architecture and systems engineering to develop a context for making rational outsourcing decisions. Section 6 revisits the make-buy decision and develops a classification scheme to aid strategic sourcing decisions based on two concepts of dependency. Section 7 provides a framework for thinking about technology dependency. Section 8 gives a few examples illustrating the ideas of the previous sections. Section 9 discusses fundamental industry dynamics that render unstable

the industry insourcing/outsourcing structures that firms attempt to construct. We present examples of this instability and a model to explain the cyclical behavior in such systems. Section 10 summarizes the main ideas in the paper and section 11 lists some open questions that need further research.

2. Manufacturing Infrastructure: Observations on Make-Buy Policies in Products and Processes

Many Japanese companies, large and not so large, write their own software to aid the design and manufacture of their products. Most US companies in the same sizes and industries buy such software. While it may be argued that this is an artifact of conditions in Japan, we believe that it is part of an important pattern in how advanced companies design and produce their products. Design software is one of many elements of what we call the *manufacturing infrastructure* to distinguish it from the technologies that comprise the products themselves.

Most companies design and make only a portion of what makes up their products, buying the rest from a complex multi-link chain of suppliers. A breakthrough in the understanding of automotive supply chains was achieved when it was found that the most successful Japanese car companies design and make as little as 30% of the components that go into their cars. [Clark and Fujimoto] For US car companies, the corresponding percentage on manufacturing outsourcing ranges from 30% at Chrysler to 70% at GM.

But there is another and less well understood supply chain, namely the one that provides the manufacturing infrastructure: hardware (machine tools, robots, and complete fabrication and assembly systems) as well as software (CAD, CAM, CAE for design as well as scheduling, logistics, and database programs for operations). An equally important and recently appreciated point is that Japanese manufacturing companies are firmly involved in the infrastructure supply chain. They make a surprisingly large fraction of their own key manufacturing equipment and write much of their design software. [Whitney 1992, 1993a] In US companies, almost exactly the opposite pattern is observed: manufacturing equipment and much of the design software is purchased from other companies. Figures 1 and 2 illustrate the pattern with the auto industry as an example.

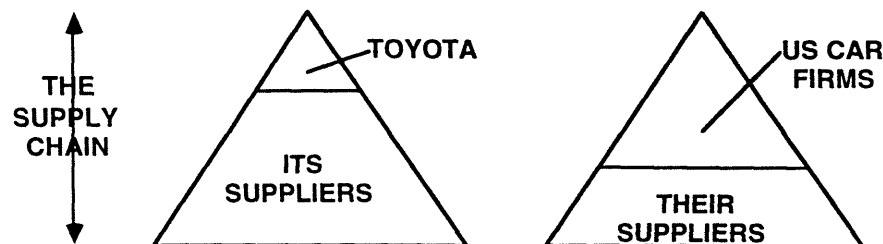


Figure 1. The Lean Paradigm Expressed as a Partition of a Triangle Representing an Industrial Supply Chain.² The triangle on the left describes Toyota while the one on the right represents typical US car manufacturers. The point here is that the chain broadens as it descends, encompassing more companies and more skills and technologies. Toyota maintains "leanness" according to this portrayal by outsourcing near the top of the chain.

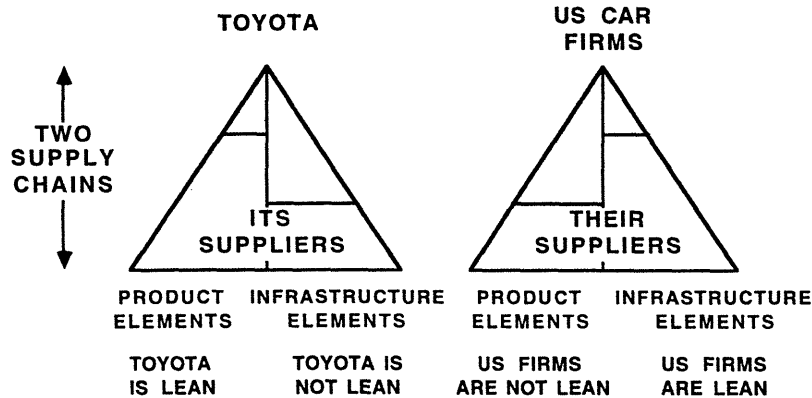


Figure 2. Lean Paradigm Restated to Distinguish Product from Infrastructure Procurement. Based on this comparison, Toyota does not appear lean on the infrastructure side while the US firms do.

Figures 1 and 2 could be generalized by the table in Figure 3. As a broad generalization about Japanese and U.S. companies, it is certainly not universally true. However, we have observed many companies that fit the pattern.

	JAPANESE COMPANY	US COMPANY
	"YOU LEARN BY TRYING, NOT BY BUYING"	"OUR BUSINESS IS CARS, NOT ROBOTS"
PRODUCT COMPONENT OR SYSTEM	<input checked="" type="checkbox"/> BUY <input type="checkbox"/>	<input type="checkbox"/> MAKE <input checked="" type="checkbox"/>
INFRASTRUCTURE COMPONENT OR SYSTEM	<input type="checkbox"/> MAKE <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> BUY <input type="checkbox"/>

Figure 3. Japanese companies buy much of their product components but make much of their infrastructure components. US companies tend to do the opposite. Japanese companies that fit this trend include Toyota, Nissan, IBM Japan, Sony, Hitachi, Matsushita, Mitsubishi, Nippondenso, Epson, and Ricoh. One might say

²This diagram is from "Japan: The Reluctant Missionary," presentation by J. Womack to participants in the MIT Japan Program. Nov 16, 1994.

that the Japanese companies are *lean* in product elements, but not in infrastructure elements, whereas the opposite is true for the U.S. firms.³

Most recently, however, many U.S. companies have begun moving from the upper right to the upper left of this table, that is, outsourcing many components as well. Chrysler is one of the widely hailed prototypes of this business model. General Motors is a holdout in the upper right cell, although not perhaps as a strategic choice but more as a necessity due to their UAW contract, illustrating one type of constraint on rapid corporate/supply chain re-engineering.

Japanese automobile firms strongly support in-house CAD development as well as that of key manufacturing equipment. Such equipment may include robots, machines that cut stamping dies, sensors used in manufacturing, and assembly equipment. Firms in consumer electronics have moved even more strongly than car firms into assembly robots and now can assemble impressively complex mechanisms very rapidly and dexterously. One example is Sony, whose robot systems assemble delicate and precise products such as video cameras. Another example is Matsushita, which developed essential technology for attaching electronic components to printed circuit boards. Both firms now sell these respective technologies and are either the only source or one of only a few that can deliver their level of flexibility, programmability, speed, and precision.

The pattern in the bottom row Figure 3 seems to apply in semi-conductors as well. Many major Japanese firms in this industry make or at least co-develop some advanced processing equipment, while most US firms buy all of theirs. For example, Toshiba's semiconductor business unit, the second largest Japanese semiconductor company, has a lithography R&D group that has about ten times as many technical experts as a similarly-sized U.S. semiconductor business unit, and makes its own advanced lithography equipment for R&D purposes in advance of the market availability of similar equipment.

Historically, the Japanese concentrated on DRAM (Dynamic Random Access Memory) design and production, exploiting their skills in precision clean manufacturing, whereas, U.S. firms focused on microprocessor design and production, reflecting their skills in software systems and logic design. Through the early 1990's microprocessor makers were able to utilize the previous generation of manufacturing technology developed for DRAMS, whose manufacturers tended to develop new processes and equipment at each new generation.⁴

³ A Japanese engineer related that "You learn by trying, not by buying." Implicit in this statement is the idea that learning itself has very high value. [Whitney, 1992]

⁴The microprocessor makers labeled this state of affairs as "drafting DRAMS," a reference to bicycle racing, where the strategic advantage falls to the racer who can stay directly behind (in the draft of) the race leader(s), conserving energy while the leader takes the full brunt of the wind drag. More recently, however, microprocessor firms such as Intel have found themselves in a position of needing new processes (e.g., more metallization layers) in advance of the DRAM industry's

In Europe, in the auto industry for example, we find a hybrid of the US and Japanese patterns in which companies are comfortable making a considerable fraction of their manufacturing hardware or at least adapting it significantly to their needs. However, they tend, like US firms, to buy software. [Whitney 1993c]

3. The Search for an Explanation: First Level of Observations

Why do these differences exist? The quotes in Figure 3 are part of the story: internal attitudes within the companies are quite different and have been for several decades. US companies want to concentrate on what they feel they do well and tend to value the product most highly. Many Japanese companies tend to view manufacturing in a holistic way. They know it is difficult to learn how to do well and they want to maintain control of as much of the process design and production chain as possible.⁵

Japanese companies operate in a different national context and historical background which may help explain why they operate this way. A quote from [Friedman and Samuels] puts it well.

Japan, we believe, values industries differently than does America. . . . [and believes] that industries have importance beyond the goods they produce. Acting on this belief, the Japanese are driven to procure or develop skills and knowledge that they may lack for their domestic economy so that non-production benefits--especially learning and diffusion--can be realized at home. Industrial policy in Japan is guided by the effort to maintain the nation's knowledge and technology base rather than to produce a specific product to which a domestic firm might affix a nameplate. . . . The U.S., in contrast, does not value industries in this way. . . . leading to wholesale capacity losses, or even domestic skill displacement from the American economy that Japan would never tolerate. . . . As we have seen in the aircraft industry, Japan is willing to pay (and pay dearly) for the same technical knowledge that the U.S. is willing to transfer abroad because it values the ancillary industrial results of that knowledge as much, or more than, the ability to make specific goods.

What advantages for individual companies are there to one strategy or the other? The product is what customers buy, not the underlying processes, so concentration on the product is not misplaced. But manufacturing skill shows up in many areas that customers notice in one way or another: quality of fit and finish, the rate at which new models come out, the time it takes for a custom order to be built and delivered, and product durability and reliability.

The history of the U.S. consumer electronics industry (see, e.g., Dertouzos et al) illustrates vividly how product skills can follow process skills. In that industry, the repeated surrender of increasingly complex electronic products (from transistor

needs (or willingness to invest). As a result, the DRAM makers no longer unequivocally drive equipment development.

⁵One Japanese manager in charge of in-house manufacturing equipment development stated that Japanese companies are careful to avoid releasing valuable technology to suppliers.

radios to black & white TV's to color TV's) from American firms to Japanese firms between the 1950's to the 1970's all began with the outsourcing of circuit board stuffing to the Japanese following the end of World War II. The Japanese electronics firms leveraged the knowledge base from assembly to component development and product development, eventually completely supplanting their original American customers.

Beyond these general statements, several other points have been repeatedly related to us over the course of many interviews and student internships.

1. Companies that design and build manufacturing equipment know what that equipment can do, allowing product development to more confidently and more readily fit to the process capability.

2. Conversely, product developers can more easily get the processes tailor-developed for the products they are designing.

3. If the product design and the equipment match from the start, then operating the factory becomes much easier because many unnecessary startup and operating problems will not occur--the operations people can concentrate on making the factory more efficient.

4. Companies that build manufacturing equipment are better at buying equipment as well: They can competently specify their needs and evaluate suppliers on capability and price.

5. Companies that build their own equipment typically understand better how to maintain it and obtain better uptime. Conversely, fear of maintenance challenges in externally procured equipment is known to have brought new technology acquisition processes to a halt in some companies.^{6,7}

6. Companies that produce their own design software can tailor it to their company culture and design procedures permitting more seamless data sharing with a family of design tools that can access, modify, check, and distribute that information.

7. Companies that mostly buy infrastructure elements face the need to fund firm-specific development projects at a supplier who may re-package proprietary knowledge for other customers or exploit its status as a sole supplier of uniquely tailored assets or servicing of those assets.

⁶Based on a series of interviews by Geoffrey Parker and Sharon Novak, MIT Sloan School Ph.D. candidates.

⁷The Japanese book Introduction to TPM (Total Productive Maintenance) [Nakajima] explicitly offers the ability to maintain the equipment as an argument in favor of developing it in-house.

8. Due to the extreme business cycle volatility (and resulting layoffs) typically experienced by equipment suppliers, customers rarely have control of how deeply the suppliers cut into the expertise that serves as the technical memory essential for supporting previously developed systems and designing new ones.⁸⁹

On the other hand, every firm bumps up against the limits to vertical integration at some point. The pitfalls of over-integrating are also many and varied:

1. Although integrating across several functions may ease coordination significantly, it may be very costly to acquire the assets required to support this activity, yielding an unacceptably low return on the investment.

2. Managing the wider variety of activities required to integrate across a wide range of needs may demand a wide range of skills which may not be particularly synergistic in terms of the management skills required to use them effectively (diseconomies of scope).

3. Acquiring key suppliers and having them focus on supporting internal needs may insulate the internal suppliers from market forces, leading to complacency in cost, quality, or technology improvement.

4. A technology source acquired due to criticality at one point in time may be made obsolete by some technological breakthrough.

5. Vertical integration increases the capital investment required, a binding constraint for some firms.

6. As industries are buffeted by the winds of creative destruction, core competencies can easily turn into core rigidities.¹⁰

These points are merely illustrative of the debate and lack of consensus on rules of thumb for make/buy decisions.

4. The Product Realization Process

To understand these issues better, we need to consider the entire product realization process for a typical manufactured product. In order for a company to be able to buy anything effectively, be it a product component or a segment of the manufacturing infrastructure, it must have an effective design process. This process

⁸See Anderson, Ed, Charles Fine, George Gilboy, and Geoffrey Parker, "Upstream Volatility in the Supply Chain: The Machine Tool Industry as a Case Study, Working paper, MIT Sloan School, May 1995.

⁹Companies at the top of the supply chain often see this positively as a chance to export layoffs. Few companies feel a need or see the advantages of nurturing and preserving their supply chain.

¹⁰Leonard-Barton, Dorothy, 1992, "Core Capabilities and Core Rigidities: A Paradox in Managing New Product Development, *Strategic Management Journal*, 13:111-125.

follows similar paths in all successful companies, and it is similar whether one is designing a product or a production system. A schematic diagram of this process appears in Figure 4.

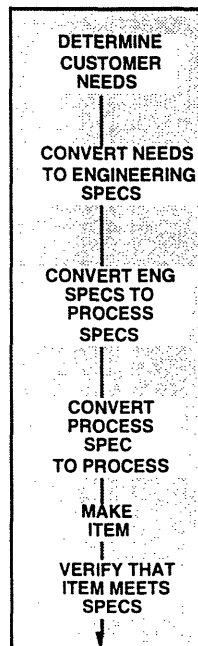


Figure 4 Simplified Schematic Diagram of the Product Realization Process. This schematic emphasizes those features of the process in which requirements are converted into successively more detailed realizations. The steps shown are repeated again and again with smaller and smaller elements of the product.

This process begins with a customer-driven statement of requirements and steadily breaks them down into sub-requirements, associating these lower levels with physical things that can fulfill them. This is called "requirements flowdown" in the aerospace industry and it must be carried out systematically with careful attention to the overall system that is desired. Each element contributes not only to the level in which it lies but to levels far above those immediately adjacent. Understanding this web of relationships is essential for good design. Knowing how to make the inevitable tradeoffs is also vital.¹¹ The ultimate arbiter of how the trades should be made is the customer at the top of the decision chain. Understanding the customer and relating his needs to very small details of the product is the most difficult part of design. Companies that do it well apparently have developed a very special set of skills. These skills comprise what is called system engineering, about which we will have much to say later in this paper.

¹¹A perennial tradeoff is that between tolerances and cost: tighter tolerances presumably increase quality but usually increase the cost. A friend of mine in the machine tool industry says that one customer demands "impossible" accuracy in the conviction that this will maximize quality even if the requested accuracy is not attained. This approach may reflect the buyer's huge financial resources, limited engineering skill, or both.

Now, it is clear that both Japanese and US companies buy at least some of what they sell, so there must come a point in the process of designing anything when a make/buy decision is made. The crucial issue is to understand what is needed and to find a competent supplier who can deliver it, perhaps one that takes on a portion of the design process as well as manufacture. If that supplier is in-house, the decision is "make." If the supplier is external, the decision is "buy." GM, Ford, or Toyota say, in effect, "We understand our needs and our customers' needs well enough that we can explain them to competent suppliers, who make and deliver what we want." When GM or Ford says that, they are usually talking about manufacturing equipment and systems. When Toyota says it, it is usually talking about car components.

In addition, at General Motors, for example, some internal suppliers have been placed organizationally at arm's length, so sourcing by GM's North American platform organization to GM's Delphi Systems is "make" at the corporate level, but "buy" at the division level.¹² Whether GM will get the best of both worlds or the worst from this system depends critically on how they manage technical capabilities and how these capabilities are made available and exploited across the corporation.

5. Product Architecture and Systems Engineering

Ulrich describes the concept of product architecture and distinguishes crisply between integral and modular product architectures. A product with a modular architecture has components that can be "mixed and matched" due to standardization of function to some degree and standardization of interfaces to an extreme degree. Home stereo equipment has a modular architecture; one can choose speakers from one company, a CD player from another, a tape deck from a third, etc, and all the parts from the different manufacturers will assemble together into a system. IBM-compatible computers are also quite modular with respect to CPU, keyboard, monitor, printer, software, etc, as are adult's bicycles.¹³

A product with an integral architecture, on the other hand, is not made up of off-the-shelf parts, but rather comprises a set of components and subsystems designed to fit with each other. Functions typically are shared by components, and components often display multiple functions. Airplanes are an example. One cannot take a wing off the shelf from one supplier, an engine from another, avionics from a third, and expect to end up with a viable (flyable) system. Rather, the product must be developed as a system and the components and subsystems defined by a design process exerted from the top down, rather than the bottom-up

¹²Clark, Fujimoto, and Ellison count outsourcing to an internal division as "buy" in their research on world car design practices. As a result, they may overstate the degree to which companies like GM actually buy from outside firms. {need citation}

¹³ Crocker, et al provide a very nice illustration of product architecture and components suppliers in the bicycle industry. {need citation}

design process that may be used by a bicycle manufacturer. Systems engineering¹⁴ is the term often used to refer to such a top-down design process.

System engineering is a product realization process best exemplified in the aerospace industry, where its top-down process is called requirements flowdown. The process conceives the product as a series of levels, with lower levels being defined in more detail or containing subsidiary components, subsystems, or single parts. The requirements defined for the lower levels support the levels above in precisely defined ways, such as providing functions, physical support, power, insulation, and so on. "Requirements flowdown" recursively determines the requirements of the level above, starting with the customer's needs, and breaks down those requirements to define the next lower level of supporting capabilities which are in turn expressed in terms of requirements that must be further broken down. At each stage in the process, the needs of the entire system are kept in view and every effort is made to avoid focusing on any one element or level in the system.

In carrying out a system engineering process, a basic decision made repeatedly is where to put the boundaries between the elements that are requested of participants or suppliers at the next lower level. This step is called "decomposition." A basic principle of system engineering is that, at each level in the process, the system or subsystem should be broken down into elements that have clear and terse interfaces with each other and with levels above them.¹⁵ Insofar as possible, complex interactions are kept within each subsystem's boundary; if necessary, subsystem boundaries are deliberately redefined precisely in order to achieve this kind of decomposition. Failure to do so courts difficulties later as subsystem designers or suppliers attempt to understand their responsibilities and how to meet them. Such failure also courts problems for the firm at the top of the process (often called the system integrator) since it becomes more difficult to keep the suppliers out of each others' way, to explain to them concisely what they must deliver, and to permit them to make their deliberations, decisions, and further decomposition's without knowing more than a few essentials about what the others are doing.

A basic skill of system engineers is thus to assess the "decomposability" of a system and to seek good ways to decompose it. In terms of the discussion above, easily decomposable items have a few well defined interfaces to the other elements in the system. It is possible to describe decomposable elements conclusively in terms of their own constituents alone, without having to describe external elements. It is important to note that not all elements or subsystems can be cleanly or conveniently decomposed. This fact influences the design process, and, as discussed below, impacts the outsourcing process as well. In effect, we will argue that the best candidates for outsourcing are those most easily decomposed. Special

¹⁴Two excellent expositions on system engineering are [Alexander] and [Rechtin].

¹⁵Interfaces are where elements and subsystems connect and across which requirements are delivered: plugs for the power, brackets for the physical support, and so on.

efforts are required when less decomposable items are outsourced or otherwise divided up and designed by separate entities.

The most decomposable items are those usually called commodities. These are available from multiple sources and at least to first order they are substitutable for each other regardless of the source. Since commodities are usually subject to large economies of scale, it is rare for a firm not to outsource them.

The Role of System Engineering Skills in Product Development and Outsourcing

An important lesson from systems engineering is that a good product design begins with a good specification and good decomposition. We assert that this is a top level skill of major importance. In fact, each stage in the system engineering process requires this set of skills, namely the ability to determine the needs of the level above (essentially a customer), break them down into supporting capabilities (decomposition), and then describe (specify) these capabilities to people or companies (essentially suppliers) who will have to figure out how to develop and deliver them. It does not matter whether these "suppliers" are members of the "customer" company or not. What matters is the following:

1. the ability to write clear and complete specifications for the needed capability
- 2a. the ability to identify suppliers capable of delivering items that meet the specifications,
- or
- 2b. the ability to create and grow suppliers capable of doing so
3. the ability to determine that the supplier has indeed delivered items that meet the specifications

It is essential to repeat that classical system engineering depends on the ability to decompose the system into subsystems cleanly at points where their interfaces are simple and clearly defined. This is a basic system engineering skill. However, the underlying technology of the system may not permit this. At such points, the system must be designed as a unit by as many people or skills as necessary to comprehend the interactions and manipulate them until a successful design is achieved. The need for this kind of close collaboration can be seen in product development arrangements that co-locate essential members of a design team. It can also be seen in reverse when companies and their suppliers fail to meet in

person often enough or depend on exchanges of merely geometric data: often the result is errors, extended design time, and so on.¹⁶

6. A Make-Buy decision framework based on system engineering concepts.

Returning to the PDP model of Figure 4, we can illustrate the range of outsourcing choices as in Figure 5, an extension of Figure 4. Here we show the most important "exit points" in the process at which a company can opt to buy rather than make, including several stages of product design and process design.

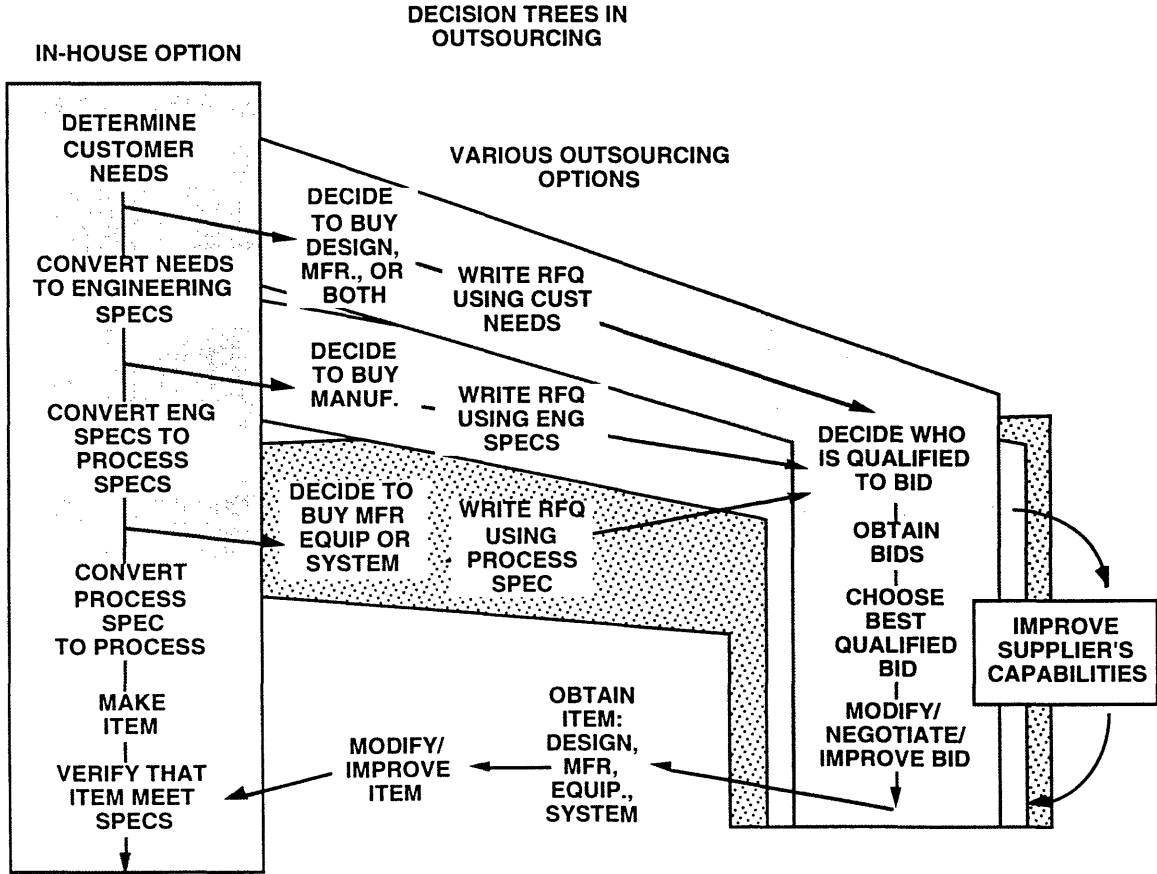


Figure 5. Illustrating Different Exit Points in the Product Realization Process Where Companies Can Opt to Buy. At the left is the schematic product realization process shown in Figure 4. When an item is outsourced, the steps at the left below the point of outsourcing must be taken over by the supplier(s). To ensure that they are taken over competently, many customer companies undertake some or all of the steps at the right.

¹⁶See work by [Eppinger et al] where a method known as the Design Structure Matrix has been used to determine the tightly linked elements of a design and to recommend which skills or constituencies should be represented on design teams.

Figure 5 suggests that companies seeking to outsource something ought to follow the set of steps listed at the far right. This list anticipates points to be made later in the paper regarding what constitutes competent outsourcing. The main assertion of Figure 5 that concerns us now is that, regardless of where a company opts to exit the in-house process, the steps it needs to follow are the same, and thus the skills it needs are the same. This statement applies whether the item being considered for outsourcing is a product item or an infrastructure item.

In fact, the process shown above is repeated countless times during development of a product. Each time a system is broken into subsystems, the designers or builders of the system may be considered the "customer" and those who design or make the subsystems may be considered the "suppliers." These suppliers may be in-house or external. In either case the same steps are required to describe the system's requirements clearly, to find sources for the subsystems that can meet these requirements, and to verify that the requirements have been met. The process keeps repeating until the subsystems finally are individual components.

Product development can thus be presented in the vocabulary of system engineering and thus requires the same skills: definition of requirements and assignment of those requirements to physical entities that may be further broken down by the same process. At each stage of breakdown, a "supplier" is sought and a competent specification of the requirements must be prepared. The more this process succeeds in subdividing the product into distinct physical items with independent functions and requirements, the more modular the design becomes. Thus the final product architecture is both a result of, and an influence upon, the success of any outsourcing.

Reasons for Outsourcing: Classes of Dependency

What are some of the reasons why a company would seek to exit the process and leave the remainder to suppliers? The classic reasons seem to be:

1. Capability: the company cannot make the item or easily acquire such a capability and must seek a supplier.
2. Manufacturing Competitiveness: the supplier has a lower cost, faster availability, etc., for what is presumably a directly substitutable item.
3. Technology: the supplier's version of the item is better for any of several possible reasons.

On the other hand, two important reasons for not seeking a supplier (and mentioned, for example, by Venkatesan¹⁷) may be termed "strategic:"

¹⁷R. Venkatesan, "Strategic Sourcing: To Make or not to Make," *Harvard Business Review*, Vol. 70, No. 6, Nov-Dec 1992, p. 101.

1. Competitive knowledge: the item is crucial to the product's performance, or the skill in producing it has been judged basic to the company's technical memory.

2. Customer visibility/market differentiation: a firm should make what matters most to the customer or what differentiates the product in the marketplace; it should buy everything else.

This list can be condensed into two main reasons why a company would seek dependency on suppliers, or equivalently into two categories of dependency:

- dependency for capacity
- dependency for knowledge¹⁸

In the former case, the company presumably could make the item in question and may indeed already do so, but for reasons of time, money, space, or management attention, chooses to extend its capacity by means of a supplier. In the latter case, the company presumably needs the item but lacks the skill to make it, and thus seeks an expert supplier to fill the gap. Between these extremes lies a range of hybrid choices, but the extremes are sufficient for our purposes to define the issues. An example of each is given next. These examples illustrate the profound differences between companies dependent merely for capacity and those dependent for knowledge. Later in the paper we will generalize these differences and recast them as categories of risk.

To illustrate these concepts, consider some of the competence/sourcing decisions made by Toyota:

<i>Toyota sourcing and strategy choices</i>	Independent for Knowledge	Dependent for Knowledge
Independent for Capacity	ENGINES	<i>RARE CASE</i>
Dependent for Capacity	TRANSMISSIONS	ELECTRONICS

Figure 6. Toyota designs, develops and manufactures virtually 100% of the engines used in its vehicles; in transmissions, Toyota designs all the products, but

¹⁸The distinction between capacity and knowledge dependence is due to Geoffrey Parker.

outsources the manufacture of 70% of the volume. Toyota depends on suppliers for design, development, and manufacture of its vehicle electronics systems.¹⁹

Toyota has often been recognized as an innovator and top performer in supply chain design and management as well as in using suppliers for "black-box" design.²⁰ The company seems to think strategically about which components and subsystems it retains inside, and which it outsources for capacity and/or design and development. Figure 6 presents some examples of the variety of practice at Toyota. Two observations are notable: First, Toyota seems to vary its practice depending on the strategic role of the component or subsystem (e.g., engines vs. transmissions). Second, some of the decisions are based on historic judgments which may be reconsidered as circumstances change. For example, Toyota has historically relied very heavily on Nippondenso for a great deal of the development and manufacture of electronic subsystems used in the vehicle. However, as electronics becomes more critical both as a percentage of the total value of the vehicle as well as to its integrality in the both design and driver interface, Toyota seems to be moving to develop more electronic competency internally.²¹

Examples of Dependency

Half Shafts: Dependent for Capacity

A purely mechanical example is provided by automobile half shafts which connect the transmission to the wheels in front wheel drive cars. Half shafts are highly-stressed, safety-critical assemblies which contain carefully engineered and precisely made constant velocity (CV) joints. A major US car firm makes half shafts for itself plus US and Japanese competitors and notes differences in the degree of oversight each such customer provides. The US customer provides basic requirements like torque and mechanical interfaces. The Japanese customer provides as much and more: a highly detailed set of test and evaluation specifications that the design must pass. A half shaft design engineer said of the Japanese customer, "They wouldn't dream of telling us how to design the CV joints but they will become very 'helpful' if our design fails any of the tests. They will lead us to find the answer they already know is right. They know because they make similar shafts themselves. I have visited them; they are the best in the world and they want us to be, too." In other words, both US and Japanese customers "depend" on their half shaft supplier, but the Japanese customer is dependent for capacity while the US customer depends in an entirely different way. It is dependent for knowledge.

¹⁹Satoshi Nakagawa, "Developing Core Technologies for Automotive Components," presentation at *Creating and Managing Corporate Technology Supply Chains*, Symposium at MIT, May 10-11, 1995.

²⁰See, e.g., Fujimoto, Takahiro, "The Origin and Evolution of the 'Black Box Parts' Practice in the Japanese Auto Industry," working paper, Tokyo University Faculty of Economics, 1994.

²¹ Personal interviews, Toyota City, May 1994.

Disk Drives: Dependent for Knowledge

The Kittyhawk disk drive was co-designed by Hewlett-Packard and a number of key suppliers in the early 1990s.²² HP set very aggressive goals for this drive in terms of very small physical size and high data storage capacity. (See Figure 7). A few years before, HP had decided that it could not afford to keep up technologically with all the elements of disk drive technology, so it developed supplier partners. HP set the main requirements for size, data storage capacity, and power consumption. These requirements in turn set the overall requirements on the main subassemblies. That is, Hewlett-Packard placed itself at the crossroads between the customer's requirements on the one hand and the decomposition of the system into components and subsystems on the other.

Once HP had determined the requirements for the key elements of the drive, namely the spindle motor, disk platters, read/write heads, head suspension arms, and electronic chips, it contracted out their design. HP designed the disk operating system software and the read/write control system. Citizen Watch in Japan was selected to manufacture the drive because of its ability to ramp up rapidly, its low margins, and its ability to manage the manufacture of complex precision equipment. The practical effect is that each element in the product was made or assembled by a different company.

We have asked several people inside and outside of HP what would prevent Citizen from assembling the same set of suppliers and going into the disk drive business itself. The answer heard most frequently from insiders was that HP's skill lies in the ability to look ahead two years and see what price and performance the market will demand, and then to convert those requirements into engineering specifications. This basically requires two skills: market knowledge and system engineering. HP's strategy was thus to control creation of the specifications and rely on suppliers to deliver most of them. HP felt safe because it was dependent only for component knowledge, not for system knowledge.

This product raises a number of fascinating questions:

- Who really has (had) the knowledge necessary to design this product, especially given the fact each critical component came from a different company?
- Of all the kinds of knowledge required - market prediction, conversion of customer needs into engineering specifications, precision mechanical design and fabrication, electro-magnetic modeling, ultra miniature electronics, precise servo controls, operating and control software, mechanical assembly, and product integration and test - which are the ones needed by HP and which could safely be allowed to wither inside and be obtained outside?

²²A more detailed description of this product and its procurement may be found in [Whitney, Nevins, De Fazio and Gustavson].

Outsourcing Described in Terms of Dependency Classes

In Figure 7 below, we present the two kinds of dependency together with a set of supporting skills required for outsourcing listed at the left. These skills rise in sophistication as one reads down the list. At the top are skills that any firm needs in order to outsource successfully, especially a firm that seeks to leverage the knowledge of others. But lower on the list are skills that require increasing in-house knowledge. A company with these additional skills can successfully outsource to expand its capacity and leverage its own knowledge. In other words, as one moves from being dependent for knowledge to being dependent for capacity, one moves from a greater degree of dependence to a lesser one. To support a move toward this position means acquiring a variety of system engineering skills.

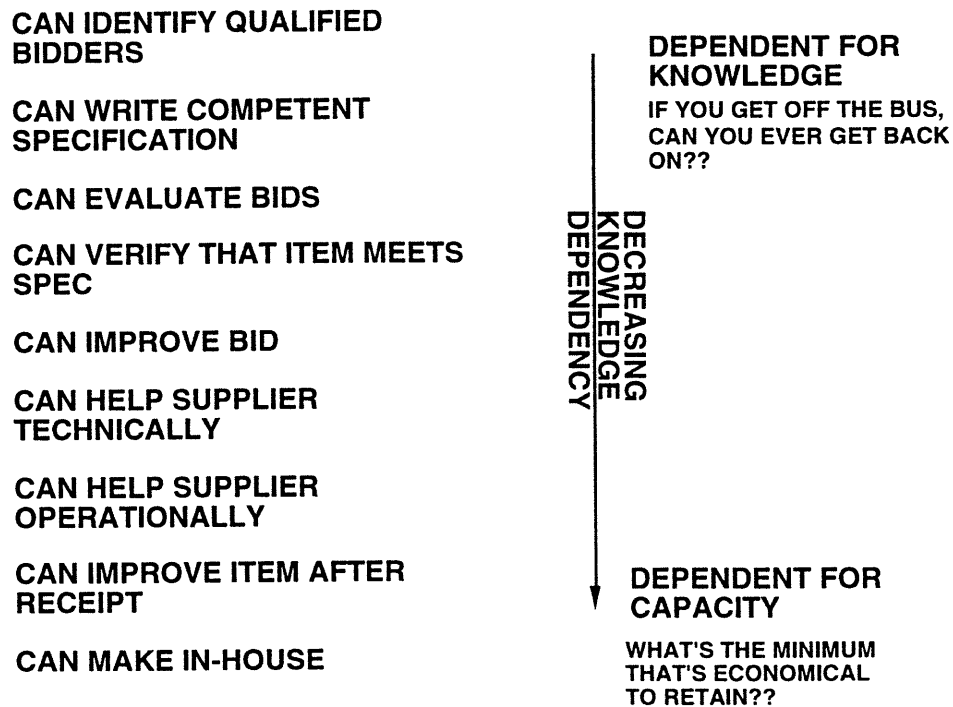


Figure 7. The Skills Required to be Dependent for Capacity Compared to Those Needed if One is Dependent for Knowledge.

The minimal skills needed are the ones at the top of the list, comprising the ability to write a competent specification, find or develop a competent supplier, and assure oneself that the specification has been met. These skills are needed regardless of whether the item is ultimately outsourced or not because they represent a recurring task in product development. The crucial question is: what other in-house capabilities are needed in order to support these minimal skills? We address this question by considering the skill sets involved with systems engineering and product development management.

Is System Engineering Toyota's Main Skill?

Based on observing Toyota in a variety of settings and relying on the work of [Ward, Christiano, Sobek and Liker] and [Fujimoto], we believe that Toyota is highly skilled at system engineering and requirements flowdown. It knows exactly what to say to different classes of suppliers making different kinds of things. It knows which outsourced items are easily decomposed and which are not, and has developed different supplier management methods for each. In order of increasing decomposability, they are commonly called "white box," "grey box," and "black box." It gives proven and trusted subsystem vendors a range of design goals rather than specific details, permitting them to do the design and production themselves. It gives unfamiliar suppliers or makers of less sophisticated components elaborately detailed drawings and specifications. [Ward, Christiano, Sobek and Liker] When it cannot find a suitable supplier, it spends years or decades developing the skills of suppliers to the extent feasible. [Fujimoto] documents several cases where it took Toyota 20 years to grow a supplier of a single item to the point where a majority of procurements of that item were of the black box type. He also shows that some items are never procured via black box methods and shows that these differences are due not to the supplier but to the non-decomposable nature of the item being procured. This provides evidence that, at least on the product side, Toyota strives to assess the degree of decomposability of the items being considered for outside purchase and has a separate procurement process for each class of decomposability. Our hypothesis is that Toyota does the same on the infrastructure side.

Given this observation, it is tempting to conclude that Toyota either cannot find enough competent or dependable infrastructure suppliers,²³ or else it has decided that it cannot, in spite of its system engineering skills, decompose the infrastructure side of the process as cleanly into buyer and seller as it can the product side.²⁴ It certainly has had plenty of time to plot its course and change it if necessary. What does this say about companies that put the in-house emphasis on product component design and production? Can they do just as well in the long run? Or is it possible that they are less sophisticated in their systems skills and simply may not be able to tell the difference between the two situations?²⁵ Of course, it may also be possible that Toyota is simply missing an opportunity to reduce costs and management load, keeping in-house a lot of work that it could give to vendors.²⁶

We do not know the answers to these questions. What we can do is look at the bottom line: Toyota is very good at design and manufacturing. They can design

²³We have been told as much by manufacturing development managers at Nippondenso, speaking of both Japanese and US suppliers.

²⁴Len Allgeier of GM put it this way: "Toyota knows where the pressure points are and avoids outsourcing them."

²⁵At one US car company it was bluntly said "CAD/CAM is not considered a core competence here. Furthermore we have no corporate level forum in which to discuss such things."

²⁶Indeed, it does sometimes appear that Japanese companies reinvent the wheel or build equipment that has no unusual technical characteristics. [Whitney, 1992] Also, Toyota appears interested in outsourcing some kinds of manufacturing equipment now that money is tight. [Personal interview July, 1994]

complex, high quality products quickly, they can ramp them up to production quickly, their first time capability is high, their ability to control costs is excellent, and they can introduce new technology into both products and processes. We also know that they pay much closer attention to various human and cultural factors than US companies generally do, and, when US companies follow similar practices, performance improvement follows.

Product Development and System Engineering Require the Same Skills

Product development and system engineering have both been discussed in this paper, and the knowledgeable reader knows or has recognized that there is considerable similarity in the two activities. The entire product development process has been described above as successive decomposition with its accompanying need to prepare specifications for the next lower level subsystems and find sources for them. That is, the skills of decomposition and preparation of specifications comprise basic strategic skills. Moreover, these skills are applicable not only to the explicit product development process itself but also to the outsourcing process.

On the basis of the foregoing examples and arguments, we assert the following important points:

1. The skills necessary to do a good job of product or process design are the same regardless of whether the end item will be made or bought; these can be condensed into the ability to write a competent specification and be sure that it will be realized.

2. These are essentially system engineering skills. That is, one must be able to define the performance requirements of each candidate subsystem and determine if those requirements can be stated clearly and independently of the requirements and performance of other subsystems. If so, then the subsystem can be decomposed, worked on separately, and, if appropriate, outsourced. If not, then it may not be a suitable candidate, and another decomposition must be sought, with new subsystem boundaries and new candidate subsystems. In terms of Ulrich's distinctions, a product with a modular architecture presents more suitable candidates or equivalently has more items suitable for outsourcing.

3. Product development is therefore an iterated process of system engineering, decomposition, and outsourcing decisions. The success of each step depends on the degree of decomposability or modularity of the items being specified and the competence with which the specifications for each item are drawn up.

4. The skills required for outsourcing are precisely those used to carry out product development and system engineering.

5. When some companies consistently outsource items that other companies consistently make in-house, it is important to find out why.

This list basically restates the content of Figure 5 in the vocabulary of system engineering.

In the next section we will try to define the risks in outsourcing. We will see that the risks depend on interactions between types of dependency and degree of decomposability.

7. The Risks Inherent in Different Kinds of Dependency

The discussion above indicates that companies cannot avoid being dependent on others to one degree or another. We have identified dependency for knowledge and dependency for capacity as the main types. In addition, we have noted that some items among product and infrastructure elements are more easily decomposed than others, and that good system engineers can recognize the difference between decomposable and non-decomposable items. The decomposable ones are more suitable candidates for outsourcing because decomposability makes it easier to carry out the first few elements at the top left of Figure 7.

The questions at the end of the HP disk drive story indicate that there are risks in outsourcing, and that these risks involve some coupling between the kind of dependency and the degree of modularity or decomposability of the items for which dependency is an issue. This coupling is illustrated in Figure 8.

MATRIX OF DEPENDENCY AND OUTSOURCING

		DEPENDENT FOR KNOWLEDGE	DEPENDENT FOR CAPACITY
OUTSOURCED ITEM IS DECOMPOSABLE	A POTENTIAL OUTSOURCING TRAP	<p>YOUR PARTNERS COULD SUPPLANT YOU. THEY HAVE AS MUCH OR MORE KNOWLEDGE AND CAN OBTAIN THE SAME ELEMENTS YOU CAN.</p>	<p>BEST OUTSOURCING OPPORTUNITY</p> <p>YOU UNDERSTAND IT, YOU CAN PLUG IT INTO YOUR PROCESS OR PRODUCT, AND IT PROBABLY CAN BE OBTAINED FROM SEVERAL SOURCES. IT PROBABLY DOES NOT REPRESENT COMPETITIVE ADVANTAGE IN AND OF ITSELF. BUYING IT MEANS YOU SAVE ATTENTION TO PUT INTO AREAS WHERE YOU HAVE COMPETITIVE ADVANTAGE, SUCH AS INTEGRATING OTHER THINGS</p>
OUTSOURCED ITEM IS INTEGRAL	WORST OUTSOURCING SITUATION	<p>YOU DON'T UNDERSTAND WHAT YOU ARE BUYING OR HOW TO INTEGRATE IT. THE RESULT COULD BE FAILURE SINCE YOU WILL SPEND SO MUCH TIME ON REWORK OR RETHINKING.</p>	<p>CAN LIVE WITH OUTSOURCING</p> <p>YOU KNOW HOW TO INTEGRATE THE ITEM SO YOU MAY RETAIN COMPETITIVE ADVANTAGE EVEN IF OTHERS HAVE ACCESS TO THE SAME ITEM.</p>

Figure 8. The Matrix of Dependency and Decomposability. Whether to outsource or not may depend on whether one seeks knowledge or capacity, and whether or not the item being sought is easily decomposed from the rest of the system.

The risks of outsourcing depend on which of the four situations in the matrix the case falls into. The top level skill to retain appears to be the one that permits a company to decide which cell in the matrix it is in at any point in the product realization process. Unlike other formulations of "core competence," which focus on product constituents, this paper asserts that the top level competence is not one of product at all but rather lies at the center of a major process.

Toyota's choice to outsource major portions of its cars but retain infrastructure capability in-house represents, in our opinion, the free choice of a skilled system engineering firm. A very plausible explanation, stated above, is that Toyota is convinced that infrastructure is less decomposable than car parts. Note that this interpretation of Toyota and other Japanese firms does not depend on an appeal to distinctive national characteristics or attitudes but rather stems from conditions imposed by the relevant technologies themselves and the attendant organizational issues involved in designing and manufacturing complex products. Toyota has

shown that it can find or develop car parts and systems suppliers. What prevents Toyota from repeating this activity in the infrastructure arena? Our answer is that it would put Toyota in the lower left corner of Figure 8: being dependent for non-decomposable infrastructure knowledge.

The formulation given here is not complete. It does not offer a formula for success and it does not identify all the risks or ways to mitigate them. As noted above, it may lead to uneconomical activities if one seeks to outsource capacity. On the other hand if one is dependent for knowledge, one may have to accept the fact that this situation is not stable. Knowledge is primarily in the heads of people who may leave the company during an ill-structured down sizing or in a move to capitalize on their knowledge in a new company. (The next subsection elaborates on this point.) Extraordinary efforts may then be needed to identify key knowledge, assess its non-decomposability, and make totally new efforts to capture it in some persistent and transferable way.

8. Examples and Discussion

Knowledge Capture and Outsourcing Decisions: Keep the Learning Opportunities In-house

One way to link outsourcing decisions and knowledge capture can be observed in companies like Toyota, Sony, Honda, and Matsushita. As noted above, these firms build many essential elements of their process infrastructure. A crucial advantage they have as a result is the ability to diffuse the knowledge behind these developments rapidly throughout the company. More broadly, they can couple these developments into their product development process, giving that development a look ahead to emerging process capabilities; and it can reap rapid feedback on how the new technology did in meeting its goals.²⁷ The "coherent system" thus consists of defining requirements for infrastructure capabilities, developing the capabilities, and learning to improve them through direct application.

Honda's procedure in the case of engine machining equipment is to try new technologies in their domestic plants for a year; if successful there, they try the new equipment in their US plants for a year; if successful there, they make the equipment available for sale. This compresses internal learning cycles and effectively keeps competitors permanently three years behind.

In Sony's case, the item being marketed consists of an assembly robot and an innovative method of preparing, feeding, and orienting the parts for the robot to pick up and assemble. Notably, the robots themselves are similar to ones made by numerous companies. The robots alone are hardly able to add value to the assembly process; it is the combination of the robot and the part feeding equipment that

²⁷This process is analogous to one cited by [Porter]: a key enabler of excellence is close contact with a sophisticated and demanding customer.

makes this a "coherent system." Polaroid chose Sony precisely because it offered this system.²⁸

More generally, Sony and Matsushita are huge companies with deep financial and technical resources. Their respective competitors in the US in the 1970s were firms like Unimation in robotics and USM in circuit board assembly. Both were small firms with limited resources, pioneers in their time but unable to move to the next generation technology. The limits they faced were not only financial and technical but also organizational: They were unable to interact early in the design process in a credible and effective way, and thus unable to influence this process or be influenced by it.

9. Dynamic Instability of Core Competencies and Industry Structure

In the setting we have presented so far, one might be inclined to conclude that firms should build strong systems engineering and product development (PD) skills, develop the appropriate product architecture according to good PD practice and customer/market needs, and then develop an outsourcing strategy consistent with perceived core competency needs and product decomposability. In a very static industry, this might be a reasonable approach, but experience suggests that many industries are quite dynamic in ways that render core competencies and product architectures highly unstable.

Consider, for example, the model of the computer industry over the past two decades articulated by Intel's CEO, Andrew Grove. From the early 1970s through the early 1980s, the industry had a strong vertical structure with each competitor offering products with fairly integral architectures. Figure 9 illustrates a sketch of the industry structure during this period when IBM was the clearly dominant firm.²⁹ Each company in this era provided all the key subsystems of a computer system, maintained internally broad technological competencies across these subsystems, and offered systems that had very little "mix and match" capability.

²⁸Interview with Norman Ward, 1989.

²⁹The vertical vs. horizontal models of the computer industry captured by Figures 9 and 10 are due to Andrew Grove of Intel. We have also been stimulated by the model of [Farrell, Hunter, and Saloner] which addresses systems competition versus component competition and also builds on Grove's model. We believe our contributions here are a fuller articulation of the dynamics between horizontal and vertical structures and the connections to sourcing strategies and core competencies.

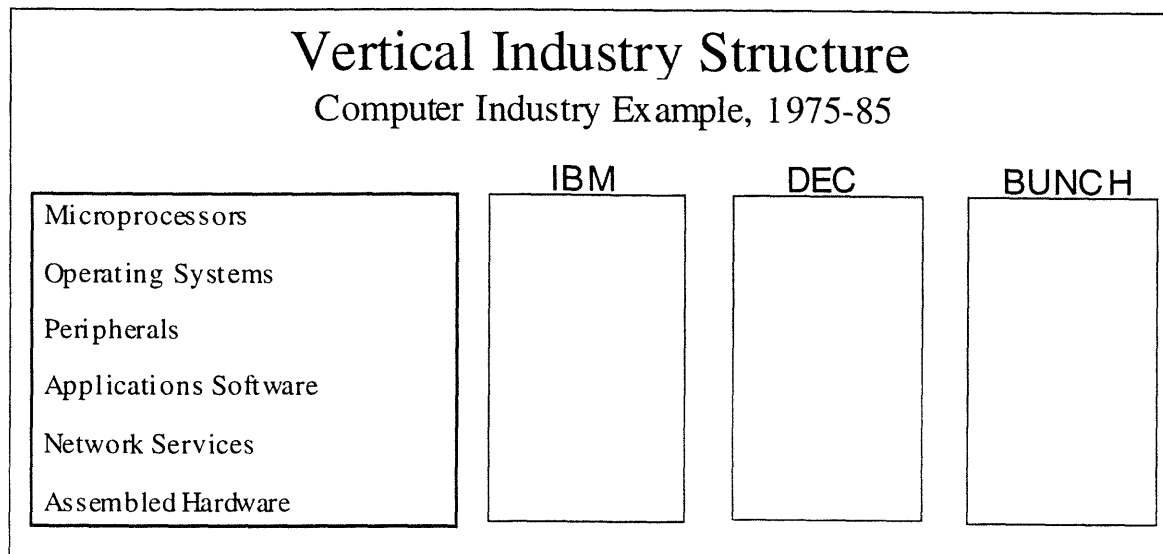


Figure 9. From the early 1970's to the early-to-mid 1980's, the computer industry was dominated by vertically-integrated systems suppliers. IBM strongly dominated virtually every aspect of the industry in this period. Its growth rates were sometime jokingly measured in "DECs per year."

Although this structure survived for some time, IBM was constantly under attack. Since it had to maintain competencies over a broad array of technologies, it was vulnerable to focused attacks on each of the many subsystems that made up the system. To maintain its position, IBM needed to keep a relatively closed architecture and offer the best "systems package" so that customers wouldn't leave them for a competitor that offered much better performance on a subset of the necessary subsystems. In the language of [Farrell, Hunter, and Saloner], the systems supplier has to be at least a "jack of all trades," if not the best in one or more subsystems. However, the precariousness of the situation for a systems supplier should be clear.

Against the backdrop of the industry structure of Figure 9, in the late 1970's IBM faced a technology supply chain decision (i.e., a simultaneous design of product and supply chain) for the launch of a product to compete with the upstart Apple II. IBM's personal computer group chose to break with tradition and use a modular architecture with the microprocessor outsourced from Intel and the operating system outsourced from Microsoft. This set of decisions catalyzed a dramatic change in the industry to a "horizontal" structure, with highly modular architectures for the dominant product ("IBM-compatible" personal computers). The modular (mix and match) architecture created significant competition in each of the "rows" of the industry illustrated in Figure 10.

Horizontal Industry Structure

Computer Industry Example, 1985-95

Microprocessors	Intel	AMD	TI	etc
Operating Systems	Microsoft	Mac	Unix	
Peripherals	HP	Canon	Samsung	etc etc
Applications Software	Microsoft	Lotus	Borland	etc
Network Services	Novell	Lotus	EDS	etc
Assembled Systems	HP	Compaq	IBM	Toshiba etc

Figure 10. Since the mid 1980's, the computer industry has been dominated by highly modular systems. With such an industry structure, competitive rivalry takes place primarily within the rows.

Competition has been quite vigorous in many of the "rows" of Figure 10. However, this structure may also prove to be quite unstable. In particular, once a firm comes to dominate its row it tends to look to how it can exploit its market power by expanding vertically. Both Microsoft and Intel, each of which came to dominate its row have exhibited this behavior. In the case of Intel, it has forward integrated into the design and assembly of "mother boards," making deep inroads into the value added typically controlled by the systems assemblers. In addition, with each new microprocessor generation, Intel has added more functions on the chip that traditionally were offered by applications software suppliers. In the case of Microsoft, dominance in operating systems has been followed by entry into compatible applications software and network services. In both these cases, the vertical integration is accompanied by a product that is moving in the direction of offering a proprietary system rather than a modular component.

Alternately, a member of a highly competitive row may find itself with low profit margins because it provides merely a commodity module in an architecture designed by someone else. This circumstance, too, can drive a firm to increase its vertical integration.

Figure 11 attempts to represent this dynamic instability by illustrating the forces that drive the cycles from vertical industry structures with integral-architecture

products to horizontal industry structures with modular-architecture products and then back to vertical again.

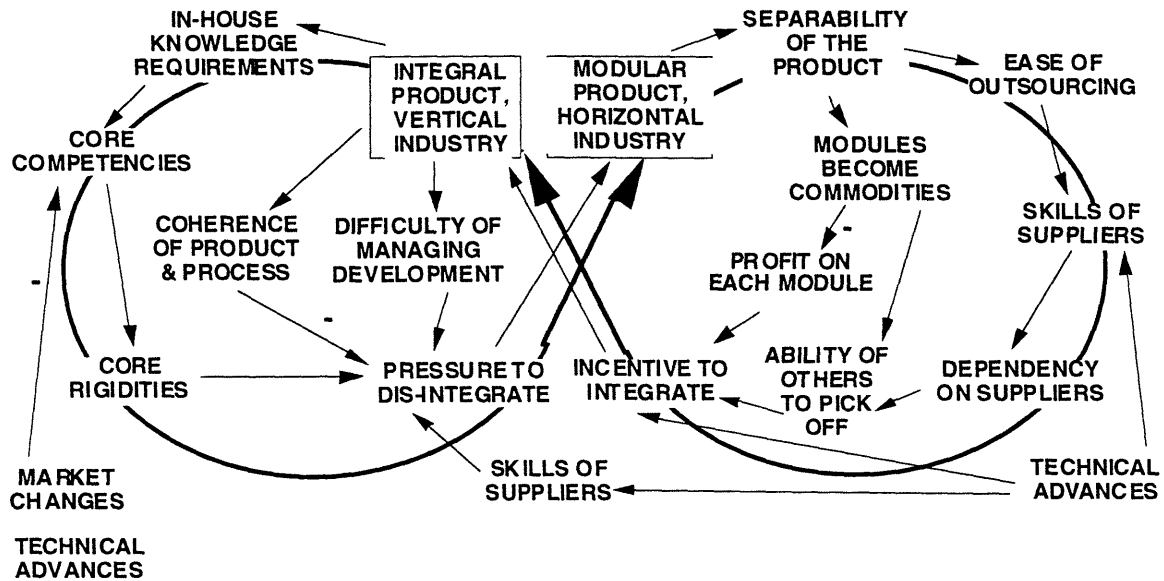


Figure 11. Influence Diagram of Integral-Modular Dynamics. (Each arrow is a reinforcing influence unless marked with a minus sign.) In an industry exhibiting a vertical structure with an integrated (systems) product, a number of forces (niche competitors, the complexity of the task of staying ahead technically with a very complex product, and the organizational rigidities [Leonard-Barton] that can set in once a firm has an established market position) push toward a loss of the established position and possible disintegration of the product architecture and industry structure. On the other hand, with a modular product and horizontal industry structure, numerous forces (technical advances, market power in one or more module suppliers, potential profitability from integrating into a proprietary system offering [Ferguson and Morris]) push toward the integration of product architecture and industry structure.

A second illustration of the dynamics of core capabilities, product architectures, and industry structure appears in the disk drive industry. [Christiansen] Christiansen traces the history of disk drives over several product generations and finds that generational breakthroughs typically require an integrated product architecture created by a vertically integrated firm, with correspondingly limited outsourcing. Within generations, components get defined and commoditized, the industry becomes more horizontal, suppliers are numerous, and outsourcing is easier.

In the face of the type of dynamic instability illustrated above, our biases are towards retaining internal technological capabilities when in doubt. History suggests that those with thin capability sets can easily get lost in the waves of creative destruction. We believe that successful companies will be deep in technical and systems capabilities, but highly flexible in their abilities to redeploy and redirect

these assets. The challenge will be to keep the core competencies from becoming rigidities, and choosing the competencies well--the make/ buy competency. The other path--staying agile by using a constantly changing stable of suppliers as needs change--seems to us extremely risky in the light of suppliers' opportunities to capture the rents when the opportunity arises (i.e., Intel and Microsoft vs. IBM).

10. Summary

This paper has presented a case that management of the outsourcing process is a core competence. The case rests on several underlying themes. One is that outsourcing is in fact an element that appears repeatedly in various guises in other important activities, the main one being the product development process itself. In that activity, outsourcing appears in the act of flowing down requirements and determining how they should be accomplished. Another underlying theme is that the skills required to do outsourcing competently are precisely the skills of system engineering. These skills duplicate the act of decomposing systems into subsystems and defining their requirements. Once a clean decomposition has been found, the item can be outsourced if desired or necessary.

In formulating our position, we sought to merge the elements of product development, systems engineering, product architecture, and supply chain management, asserting that the skills required to manage these are similar and constitute a major core competence.

We also argued that outsourcing creates two different kinds of dependency, the least risky being dependency for capacity when one retains the knowledge, the most risky being dependency for that knowledge. This range of risks is moderated by the degree to which the item for which one is dependent is decomposable from other items or activities.

Third, we gave examples indicating that product development and manufacturing infrastructure are difficult to decompose because their tools and equipment are necessarily tightly linked to elements of the product, to key processes and learning activities in the firm, and to each other. Outsourcing elements of these processes is therefore particularly risky and requires the most attention and deliberation. The ability to decompose particular elements of either of these processes implies nothing about the decomposability of the processes as a whole or of other elements.

Finally, we showed that the structure of the product and that of the industry can be quite similar; both may tend to be integral/vertical or modular/horizontal at any one time. More importantly, these configurations appear to be unstable for a variety of related technical and economic reasons, and have been found in several industries to cycle from one form to the other and back.

In coming to these conclusions, we relied heavily on evidence that Japanese manufacturing firms make these choices in quite the opposite way from US

companies. Japanese companies appear more sensitive to processes, continuity, and learning than their US counterparts, and this sensitivity appears to be reflected in their behavior in this arena. However, examples from many industries and countries appear to bolster the views expressed here.

11. Open Questions

This paper is based on observation, speculation, and extrapolation of current academic research. It remains to be seen how much of it is true. Some of the questions that need to be addressed are:

1. What are the detailed policies and processes by which companies make individual technology choices, such as make/buy of equipment or software? What would detailed case studies at Toyota, GM, Chrysler, and others reveal?

2. How do companies go about evaluating potential suppliers? How do they evaluate different bids for the same thing? To what degree are economic, systems, and architecture issues acknowledged and reflected in the process?

3. How do firms go about formulating the specifications on which equipment or software suppliers will bid? How are the needs of the product designers reflected in the specifications? Or the needs of the factory's operating staff? Are these processes different, in timing or content, if the supplier is in-house rather than a different company?

4. How do companies decide what capabilities are "core competencies" that must be kept and nurtured in-house? What is the status of product competencies as compared to infrastructure competencies? Are the decision processes and evaluation criteria for these different competency domains similar or different?

5. How long did it take a company like Toyota or Nippondenso to develop the people, policies, skills, product-process design methods, and computer tools that place them where they are now? In the case of Nippondenso, it appears to have been a 25+ year process. [Whitney, 1993b]

6. What is the history of development of infrastructure procurement patterns in the US, Japan, and Europe? Apparently Japan and Europe have not developed infrastructure industries in either hardware or software to the extent that the US has. Why did this happen? US companies have long had the luxury of a supplier base while companies in the other regions had to develop in-house skills.

7. What engineering and management practices do companies need in order to improve their manufacturing infrastructure?

8. Can quantitative metrics be generated that will help companies assess the options of having more control over the manufacturing infrastructure?

9. What government policies or programs might encourage more investment in manufacturing infrastructure development, either in-house or at vendors? For example, Japan has provided support for leasing companies that make equipment, software, and training available to small businesses. Japan also has Prefectural Manufacturing Field Stations (like our Agricultural Field Stations for introducing new methods to farmers).

References and Bibliography

[Alexander] Alexander, Christopher, Notes on the Synthesis of Form, Cambridge: Harvard University Press, 1964

[Clark and Fujimoto] Clark, Kim, and Fujimoto, Takahiro, "Overlapping Problem-solving in Product Development," in K. Ferdows, ed., Managing International Manufacturing, Amsterdam: North Holland, 1989. See also Chapter 5 of Womack, Jones, and Roos, The Machine That Changed the World, New York: Harper Perennial, 1991.

[Christensen] Christensen, Clayton, "The Drivers of Vertical Disintegration," Harvard Business School Working Paper, Oct. 8, 1994.

[Dertouzos et al] Dertouzos, Michael, Richard Lester and Robert Solow, *Made in America*, MIT Press, 1989.

[Eppinger et al] Eppinger, Steven, and Pimmler, Thomas, "Integration Analysis of Product Decomposition's," ASME Design Theory and Methodology Conf, Minneapolis, Sept, 1994.

[Friedman and Samuels] Friedman, David and Samuels, Richard, How to Succeed Without Really Flying: The Japanese Aircraft Industry and Japan's Technology Ideology, MIT Japan Program Working Paper 92-01, presented at NBER Conference on Japan and the US in Pacific Asia, San Diego, April, 1992.

[Farrell, Hunter, and Saloner] Farrell, Joseph, Hunter K. Monroe, and Garth Saloner, "The Vertical Organization of Industry: Systems Competition vs. Component Competition," working paper, Stanford Business School, November 1994.

[Fujimoto] Fujimoto, Takahiro, "The Origin and Evolution of the "Black Box Parts" Practice in the Japanese Auto Industry," University of Tokyo Faculty of Economics working paper 94-F-1, March, 1994.

[Goldman, Nagel and Preiss] Goldman, Steve, Nagel, Roger, and Preiss, Kenneth, Agile Competitors and Virtual Organizations, New York: Van Nostrand Reinhold, 1995

[Hammer and Champy] Hammer, Michael, and Champy, James, Re-engineering the Corporation, New York: Harper Business, 1993.

[Henderson and Clark] Henderson, Rebecca. and Clark, Kim., "Architectural Innovation: The Reconfiguration of Existing Systems and the Failure of Established Firms," Administrative Science Quarterly, v 35, March, 1990, pp 9-30.

[Iansiti] Iansiti, Marco, "Real World R&D," Harvard Business Review, May-June 1993, pp. 138-147.

[Kumpe and Bolwijn] "Manufacturing: The New Case for Vertical Integration," Harvard Business Review, March-April 1988, pp. 75-81

[JTEC] "Electronic Packaging in Japan,"

[Nakajima] Nakajima, Seiichi, Introduction to TPM : Total Productive Maintenance, Cambridge, Mass.: Productivity Press, 1988.

[Porter] Porter, Michael, "The Competitive Advantage of Nations," Harvard Business Review, March-April, 1990, pp 73-95.

[Prahalad and Hamel], Prahalad, C. V., and Hamel, Gary, "The Core Competence of the Corporation," Harvard Business Review, May-June 1990, pp. 79-91.

[Rechtin] Rechtin, Eberhard, System Architecting, New York: Prentice-Hall, 1991.

[Vanketesan] "Strategic Sourcing," Harvard Business Review, Nov.-Dec 1992, pp. 98-107.

[Ward, Christiano, Sobek and Liker] Ward, Allen, Christiano, John, Sobek, Durward, and Liker, Jeffrey, "Toyota, Concurrent Engineering, and Set-Based Design," to appear in Engineering in Japan: Organization and Technology, eds. J. K. Liker, J. E. Ettl, and J. C. Campbell, Oxford University Press, 1995.

[Whitney, 1992] Whitney, D. E., "State of the Art in Japanese CAD Methodologies for Mechanical Products - Industrial Practice and University Research," Office of Naval Research Asian Office Scientific Information Bulletin, vol 17 no 1 Jan-Mar 1992 pp 89-111.]

[Whitney 1993a] Whitney, Daniel E., "Integrated Design and Manufacturing in Japan," Manufacturing Review, v 6 no 4, Dec., 1993, pp. 329-342.

[Whitney 1993b] Whitney, D. E., "Nippondenso Co. Ltd: A Case Study of Strategic Product Design," Research in Engineering Design, #5, Dec, 1993, pp 1-20.

[Whitney 1993c] Whitney, D. E., "University Research and Industrial Practice in Electro-mechanical Design in Europe - Results of a Six Month on-site Study," Office of Naval Research European Office European Science Notes Information Bulletin v 93 no 1, 1993, pp 1-52

[Whitney, Nevins, De Fazio, and Gustavson] Whitney, D. E., Nevins, J. L., De Fazio, T. L., and Gustavson, R. E., Problems and Issues in Design and Manufacture of Complex Electro-mechanical Systems, Draper Laboratory Report R- 2577, Dec, 1993.