6 Product and process architectures in the management of knowledge resources

Ron Sanchez

Introduction

In both theory and practice, effective management of knowledge resources is increasingly recognised as critical to an organisation's ability to derive economic value from its uses of both tangible and intangible resources. Because no one can realistically hope to manage well that which one does not understand in at least fundamental respects, the task of managing knowledge resources begins with the challenge of developing a conceptual framework for identifying and defining the specific knowledge resources that are important in helping a given organisation carry out its 'strategic intent' (Hamel 1989a). Given some kind of framework that enables identification of the strategically useful knowledge resources available to an organisation, the next challenge is to develop effective approaches to using existing knowledge in leveraging current organisational competences and to creating and acquiring new knowledge in building new competences (Sanchez and Heene 1996a, 1996b). The challenge of creating more effective management systems for identifying, acquiring or accessing, coordinating and augmenting strategically important knowledge is thus an essential dimension of the 'contest between managerial cognitions' that characterises competence-based competition (Sanchez *et al.* 1996).

Foss and Eriksen (1995) have characterised technology as consisting of both firm-specific capabilities and non-proprietary 'industry capabilities'. In the vocabulary of competence-based strategic management, technology may therefore be either a 'firm-specific' or a 'firm-addressable' resource (Sanchez *et al.* 1996). Managing technology-based knowledge resources requires making some difficult but strategically important decisions about where to position a firm in an 'external resources and capabilities space' (Foss and Eriksen 1995: 45). A firm chooses a resource and capability position close to those of other firms when it pursues technology and product strategies that access employees and knowledge that exist as a firm-addressable 'industry capability' because other firms are also using similar resources. Conversely, a firm may distance itself in resources and capability space by staking out a technology strategy and product strategy that requires idiosyncratic, firm-specific technology resources (or at least resources that are not usually available to

competing firms). Within this framework, a close positioning of firms in the external resources and capabilities space may bring certain benefits in the form of network externalities (discussed below), but the potential to earn rents within an industry depends on finding some dimension(s) in which a firm can distance itself from its competitors by developing and using firm-specific resources and capabilities.

Some important benefits may accrue to firms that are participants in an industry or a cluster of firms within an industry that share technology resources. Saxonian (1991) and others have argued that use of common technologies may achieve sufficient scale within an industry that firms can begin to specialise in various aspects of a technology, leading to greater technological efficiencies through specialisation of labour. In addition, users of common technologies often constitute a technology network that facilitates 'information exchange and joint problem solving between firms' (Saxonian 1991: 410), a notion which reflects Dosi's (1982) observation of the use of standardised methods for solving technological problems within a 'technological regime' (Nelson and Winter 1982). Farrell and Saloner (1985) have also suggested that the use of standards within a technological regime appear to accelerate technological progress in an industry using that regime.

This chapter seeks to extend these conceptualisations about technology as both a firm-specific and a firm-addressable resource by investigating the roles of *product and process architectures* in the strategic management of technology-based knowledge resources. The following discussion proposes that both the products which an organisation creates and the processes which it follows in creating, producing and supporting its products have *architectures* which greatly influence the structures of much of the organisation's strategically useful knowledge about technologies and markets. Clearly defining an organisation's product and process architectures is argued to be a critical step both in clarifying the nature of the technological knowledge resources an organisation has or has access to, and in delineating the relationships between an organisation's knowledge resources – which I refer to as an organisation's technological *knowledge architecture* (Sanchez 1996a). I also suggest that alternative product and process architectures an organisation may adopt have greatly different potentials for enabling identification of both existing organisational knowledge resources and opportunities for building strategically important new knowledge. In particular, the use of *modularity* in creating product and process architectures is argued to lead to greater clarity in identifying key current organisational knowledge resources and to more effective targeting of strategically useful organisational learning.

Some of the major implications of introducing the concepts of product, process and knowledge architectures into an analysis of technology resources are these:

1 Adding an architectural dimension to our characterisation of technology brings into focus the *structure* of a technology regime and the forms in

which technological specialisation can occur within that structure. The increasing use of standard product, process and knowledge architectures by a group of firms creates opportunities for firms to specialise in developing and producing *standardised components* for products, in providing *standardised processes* as manufacturing and product support services, and in applying *standardised expertise* in solving problems that arise in developing new products and processes within industry architectures (Sanchez and Mahoney 1996).

2 The theory of competence-based competition has sought to move beyond representations of resources in which the value of a resource is treated as independent of its context of use. The competence perspective emphasises that the economic value of resources can only be realised when they are *coordinated* in a given use. Thus, value may be created (or rents generated) not just through the possession of superior resources, but through superior coordination and targeting of resources (Sanchez *et al.* 1996). Standard product, process and knowledge architectures used in an industry create *information structures* that, in effect, *embed coordination* of industry participants' uses of resources (Sanchez 1996a) by defining standard inputs and outputs for components in product designs, standard processes for manufacture and support, and standard knowledge applied to problem solving. Thus, in addition to the gains in efficiencies that can result from the *specialisation* of technological resources within a technological regime, the adoption of industry standard architectures for products, process and knowledge can lead to gains in efficiencies by reducing the costs and difficulties of *coordinating* the use of specialised technological resources.

3 As Foss and Eriksen (1995: 59) have suggested, 'some measure of coordination of asset-accumulation processes often takes place, to the benefit of participating firms' when technology development within an industry leads to standardisation. As this chapter proposes, standard – and especially *modular* (Langlois and Robertson 1992) – product and process architectures provide frameworks that may help a firm to clarify its knowledge about products and processes, and thereby help to identify opportunities for acquiring or building ('accumulating') useful new knowledge resources.

4 If firms have differing abilities to use product and process architectures to define opportunities to leverage existing technological knowledge more effectively and to identify and acquire strategically valuable new learning, then a firm's *ability to use and create knowledge resources* within the framework of industry standard architectures becomes an important dynamic dimension to be added to the industry resources and capabilities space suggested by Foss and Eriksen (1995). In effect, a firm's potential for generating rents will be determined not solely by the position it chooses in the resources and capabilities space, but also by its differential ability to use and create technological knowledge in whatever position it occupies.

This suggests that the choice to be made in the resources and capabilities space is not one in which the benefits of close positioning must be strictly traded off against the benefits of distancing. A firm that has a superior ability to learn within an industry standard architecture shared with other firms, for example, may be able to reap the efficiency benefits of using industry-level knowledge and other resources that result from close positioning in the shared architecture, but at the same time may be able to generate rents by accumulating firm-specific knowledge about how to use (coordinate and target) resources more effectively within an industry architecture. In this way, architectures may function as a key mechanism for coordinating interactions between firm-addressable resources available from the technology regimes of industries and internally developed, firm-specific technology resources.

5 Product and process architectures provide a framework that clarifies the ways in which organisational knowledge and learning can occur in three modes: *know-how*, *know-why* and *know-what* (Sanchez 1996a). Analysis of the architectures associated with technology regimes therefore provides a framework for finer characterisation of firms' differential abilities to use and acquire various forms of knowledge, and for identifying different kinds of organisational learning strategies.

6 Competing product and process architectures within industries may have inherently greater or lesser efficiencies as coordination mechanisms. Comparisons of architectures may therefore help to explain differential performance levels among groups of closely positioned firms (strategic groups). For example, one could compare the performance of firms using the Windows product architecture with the performance of firms using the Macintosh product architecture. Further, since firms in one product market also compete against firms offering substitute products, the relative efficiencies or inefficiencies of the product and process architectures in common use in competing industries may help to explain differential performance levels of those industries. For example, the relative performance of competing energy industries may result in part from the relative efficiencies of the industry architectures for providing electricity vs. architectures for providing natural gas. In such comparisons, performance may be evaluated in both *static* and *dynamic* efficiency terms (i.e., architectural efficiencies in *using* and in *creating* new resources).

This discussion is organised in the following way.

The first section defines *product architectures* and distinguishes modular product architectures from other kinds of product architectures.

The next section defines *process architectures* and, drawing on the notion that 'products design organisations' (Sanchez and Mahoney 1996), elaborates the ways in which product architectures used within an organisation constrain the feasible process architectures an organisation can adopt in creating and producing its products. The discussion then explains how use of modular product architectures enables adoption of modular process architectures.

The third section explains the ways in which organisational knowledge becomes structured around product and process architectures. The clarity with which an organisation can determine the current states of its component-level and architectural-level knowledge within its knowledge structures determines an organisation's ability to identify opportunities to develop specific new knowledge resources of strategic value to the organisation. By virtue of their greater decomposition into 'loosely coupled' systems of components, both modular product architectures and modular process architectures make possible more loosely coupled *knowledge domains,* which in turn makes possible more precise identification of specific knowledge resources at both component and architectural levels (Sanchez and Mahoney 1996). The improved ability to distinguish current states of knowledge and their modes of interaction within modular architectures provides a frame of reference which facilitates greater precision in identifying opportunities for acquiring strategically valuable new forms of knowledge – i.e., for more effective strategic targeting of organisational learning.

The fourth section proposes that organisations have different kinds of knowledge within their knowledge architectures and suggests how adoption of modular product and process architectures improves the ability of an organisation to leverage and create *know-how*, *know-why* and *know-what* forms of knowledge (Sanchez 1996a, 1996b). Consequently, organising knowledge resources within a modular architecture may be a superior approach to mediating the technological and market uncertainties facing organisations, by enhancing organisational learning capabilities.

The fifth section concludes with some observations on the increasing prevalence of modular product and process architectures in rapidly evolving areas of knowledge. I suggest that one explanation for this observed association is that modular knowledge architectures create substantial positive network externalities that act to accelerate learning processes within architectural alliances of firms or within industries. However, since the creation and use of modular architectural infrastructures that accelerate learning are often 'taken for granted' by practitioners in rapidly evolving areas of technology and market development, the critical role of modularity in facilitating both firm-level learning processes and industry-level technological progress is often overlooked by management researchers interested in understanding knowledge resources and the processes that create these key resources.

1 Product architectures

Most products – including service and software products as well as assembled goods – perform several discrete functions that combine together to provide the set of functionalities that distinguish one product from another in the marketplace. After determining the desired package of functionalities or 'bundles of

product attributes' (Bogner and Thomas 1996) to be provided by a new product, product designers create a new product design by decomposing the desired set of product functionalities into a system of functional components whose individual functions collectively interact to provide the overall functionalities desired in the product (Sanchez 1996b).

A *product architecture* is created by (a) decomposing a product design into a system of functional components (Henderson and Clark 1990), and (b) fully specifying how individual components will interact with other components in that system of components (Sanchez 1995; Sanchez and Mahoney 1996).[1] The component interface specifications in a product architecture define, for example, how one component may be physically connected to another (the attachment interface), how power is to be transferred between components (the transfer interface), how signals will be exchanged between components (control and communication interfaces), the spatial location and volume a component may occupy (spatial interfaces), and various ways in which the functioning of one component may generate heat, magnetic fields or other environmental effects that must be accommodated by other components (environmental interfaces) (Sanchez 1994). In this sense, a product architecture is a more foundational concept in technology than a 'product family' targeted at a market segment (Meyer and Utterback 1993), because some kinds of product architectures enable families of products to be leveraged from a single architecture, while others do not. Architectures are also broader than 'platforms' targeted at a market niche (Meyer and Utterback 1993), because they may enable leveraging of product variations for several niches or even several market segments (Sanchez 1996b; Sanchez and Mahoney 1996; Sanderson and Uzumeri 1997).

There are two fundamentally different architectural approaches to defining component interfaces in a product design, which will be distinguished here as *conventional* and *modular* approaches to creating product architectures. The essential differences between conventional and modular approaches to defining, designing and developing new product architectures are summarised in Table 6.1.

As suggested in Table 6.1, the conventional product design process usually relies on extensive marketing research to suggest the specific product functionalities, performance levels and maximum price that will appeal most broadly to a targeted set of customers. Given a set of marketing-determined optimal product attributes, the objective of product designers is to create an optimal product design that provides the desired product attributes at the lowest possible cost or the highest possible level of product performance within a specified cost constraint. The product architectures created by a conventional product design process are typically complex designs in which technically separable functional components have been combined into integrated assemblies of components to increase performance and/or to lower costs. Development of such designs is often time-consuming and costly, however, because making changes in individual component designs during

Table 6.1 Comparison of product definition, design and development in creating conventional versus modular product architecture

	Definition	*Design*	*Development*
Conventional product architecture	'Optimal' attributes of product are determined by marketing research.	Desired functionality is decomposed into functional components, but component interfaces are specified during component development.	Component designs and product design co-evolve in a recursive process. Product architecture is defined by final component interface specifications, i.e., as the output of the development process.
Modular product architecture	Product architecture is created as a vehicle for leveraging product variations and upgraded models.	At the beginning of component development, fully specified component interfaces define input and output requirements for component designs.	Component development processes may be concurrent, autonomous and distributed, since product architecture defined at design stage does not change during development.

development to improve the overall product design may require compensating changes in the designs of many closely interrelated or 'tightly coupled' (Orton and Weick 1990) components.

In contrast to conventional product architectures, a modular product architecture is intentionally designed to permit the 'substitution' of different versions of functional components (Garud and Kumaraswamy 1993) for the purpose of creating product variations with different bundles of functionalities, features and performance levels. In effect, the task of modular product design is to create a flexible product architecture or 'platform design' (Wheelwright and Sasser 1989; Sanderson and Uzumeri 1997) for introducing component variations that create product variations for serving a range of market requirements. To create a *flexible product architecture* for leveraging product variety (Sanchez 1995), modular product design follows a discipline of avoiding 'designing in' tight interdependencies among component designs and instead seeks to create loosely coupled component designs in which any design variation in one type of component (within a specified range of variation) will work in conjunction with any design variation in another type of component (within its specified range of variation). The most familiar example of this property of modular architectures is probably the personal computer, which generally allows 'plug and play' system configuration of various hard disk drives, memory cards, monitors, etc.

Although the focus in this chapter is on the impacts of product architectures on the knowledge structures an organisation develops, it is worth noting here that the 'designed-in' flexibility of modular product architectures has stimulated their adoption in a growing number of industries where their flexibilities are being put to several strategically important uses (Sanchez 1995, 1996a, 1996b):

- *Increased product variety*. The ability to 'mix-and-match' components in a modular product architecture enables leveraging of greater product variety than can be obtained from a conventional product architecture. In addition to the familiar example of the personal computer, this property of modular product architectures is now being used strategically to offer expanded ranges of product variations in industries as diverse as automobiles, consumer electronics, home appliances, object-orientated software, banking services and power tools (Langlois and Robertson 1992; Sanchez 1994; Sanchez and Mahoney 1996).
- *Faster introductions of upgraded products*. Modular product architectures may be designed to accommodate both currently available components and technologically improved components that are expected to become available within the intended lifetime of the product architecture. The ability to introduce better components directly into a modular product architecture without having to make design changes in the product architecture enables the speedy introduction of upgraded product models as soon as improved components become available. As a result, competitive

product strategies driven by rapid product improvements become possible for firms that are able to create robust modular product architectures which let them introduce new components directly into their existing product architectures. Sony's use of a modular product architecture for its introduction of the HandyCam video camera, for example, enabled Sony to introduce four upgraded models within a 22-month period by quickly incorporating improved components into the HandyCam product architecture (Sanchez and Sudharshan 1993).

- *Lower design, production, distribution and service costs.* Leveraging several product variations from a single modular product architecture may substantially reduce design costs per product variation introduced. Economies of scale can also be realised when 'common components' that can be used across product variations are produced in large numbers. When the component variations that differentiate an individual product model can be added at a 'late point' in the production or distribution process, the reduced variety of parts used in product assembly and inventories reduce production and distribution costs per product variation introduced. General Electric's different stove models, for example, are differentiated by cooktops and control panels added in the last stages of the assembly process, while high-speed production lines assemble the common components used in all GE's stove models. Taking this concept one step further, Hewlett-Packard differentiates its ink-jet printers by adding a power supply to suit local electrical requirements and a local language instruction manual in its regional distribution centres around the world (Lee *et al.* 1993). Modular product architectures may also significantly change product service requirements by enabling the 'design-in' of self-diagnostic capabilities and easily replaced modular components that radically lower field service costs.

2 Process architectures

In its most basic sense, a *system* is a structure of parts that interact with each other to some degree. Simon (1981) argues that *hierarchical decomposability*[2] is an organising principle of complex systems, whether the system be a product design or an organisation design. Henderson and Clark (1990) and others have observed that organisations tend to organise their knowledge creation and application processes in groups whose activities are focused on the functional components in their product designs. In other words, the structure of the decomposition of component parts within a product design will tend to be reflected in the structure of the decomposition of the activities of the organisation that creates and produces the product. Adding the observation that the specific tasks that *must* be performed in developing and producing a product are largely determined by the specific product architecture and component designs a firm adopts, Sanchez and Mahoney (1994, 1996) propose that although organisations ostensibly design products, in a fundamental sense 'products design organisations'. They further suggest that the use of conventional vs. modular product architectures within an organisation imposes very different sets of constraints on the feasible organisational structures – or *process architectures* – an organisation can adopt in creating and producing its products.

Product designs can be fundamentally distinguished by the degree to which the architecture of each product has been decomposed into tightly coupled versus loosely coupled (Orton and Weick 1990) components.[3] Conventional engineering design methodology for optimising product designs to meet defined market preferences usually results in product designs composed largely of tightly coupled components combined in integrated assemblies to improve performance or reduce costs. As a consequence, *processes* for developing products composed of tightly coupled components typically require intensive communication and coordination across all component development groups, since a change in the design of one component during development may require extensive compensating changes in many other components being developed by other groups. As a further consequence, product designs composed of tightly coupled components will typically require tightly coupled development processes, in the sense that continual exercise of overt managerial authority will be needed to coordinate and adjudicate across highly interdependent component development processes. The exercise of overt managerial authority, in turn, will typically require an authority hierarchy of the sort found within the boundaries of a single firm or within a quasi-integrated group of suppliers of the sort often observed in Japan (Sanchez and Mahoney 1996). Tightly coupled component designs may also necessitate tightly coupled production and service activities that must be carried out within a single authority hierarchy.

A modular product architecture, in contrast, achieves the loose coupling of components by specifying the input and output interfaces between components to allow for a range of component variations to be used within the product architecture. Moreover, when these 'flexible' modular component interface specifications are *standardised* – i.e., not allowed to vary over some intended period of time – they create a stable *information structure* of required component input and output specifications (Sanchez and Mahoney 1996). The stable information structure of the component interface specifications in a modular product architecture provides a vehicle for *embedded coordination* of loosely coupled component development (and possibly production and service) processes (Sanchez 1995), because the component development task of one group is not affected by decisions made in another development group, as long as all development groups conform to the input and output requirements for components specified in the modular product architecture. Thus, modular product architectures make possible *modular process architectures* in which development tasks can be carried out *concurrently and autonomously* by potentially widely distributed networks of development resources, without need for continual exercise of managerial authority.

3 Knowledge architectures

Taking an information processing perspective, Malone, Yates and Benjamin (1987) propose that the difficulty of *coordinating* processes for creating and producing a product varies directly with the complexity of the description of the product. Adopting an organisational learning perspective, the discussion in this section makes an analogous argument that the difficulty of *learning* about components and their interactions increases with the complexity of the descriptions of components and their interactions within a product architecture. From this perspective, product architectures with complexly interrelated, tightly coupled components limit the ability of an organisation to distinguish clear cause-and-effect relationships within and between individual components, thereby limiting an organisation's ability to develop component-level and architectural-level knowledge about component behaviours. Further, a limited ability to define clearly what it does and does not know about component behaviours and interactions also limits an organisation's ability to identify specific new kinds of knowledge about components and their interactions which could be of strategic value to its product creation capabilities. Similarly, an organisation is limited in its ability to learn about its process capabilities when those capabilities are embedded in complexly interdependent activities in its process architecture.

By virtue of their decomposition into loosely coupled components and activities, however, modular product and process architectures create loosely coupled *knowledge domains* which both facilitate learning about specific components or activities within those domains and make possible greater precision in identifying an organisation's knowledge at both component and architectural levels of understanding. An improved ability to distinguish current states of component and architectural knowledge improves the ability of an organisation to perceive and define opportunities for acquiring strategically valuable new forms of knowledge, enabling more effective targeting of organisational learning.

To understand these interrelationships of product and process architectures and organisational learning more fully, it is useful to recall von Hippel's (1994) finding that complex problem-solving processes in organisations create 'sticky information' that is difficult to define precisely or to extract from the 'locus of problem-solving' in which it is embedded. One explanation why some forms of learning create 'sticky information' is that problem-solving involving many complexly interdependent variables creates learning that is likely to be highly context-specific – and therefore difficult to articulate in more general terms that would enable its application to other contexts. To the extent that an organisation's problem-solving leads to context-specific learning rather than more generalised learning that can be applied to other contexts, however, its knowledge resources are more limited in their strategic value (Sanchez 1997), and, recognising this, an organisation may tend not to deviate from the set of circumstances in which its 'sticky' knowledge is embedded. Thus, complex product and process architectures with tightly coupled components and activities are likely to generate context-specific 'sticky information' whose value as a knowledge resource in other contexts may be very difficult to discern, including the applicability of knowledge gained in the context of one project to other product development projects within the same organisation.

The way in which modular product and process architectures create *knowledge architectures* (Sanchez 1996a) composed of loosely coupled knowledge domains is suggested by the model in Figure 6.1. At the centre of Figure 6.1 is a modular product architecture represented by the system of loosely coupled components whose ranges of permissible interactions (represented by the arrows linking components) are fully specified by a modular product architecture. Within an organisation, these specifications of permissible component interactions may be articulated (Sanchez 1997) in the form of printed 'interface documents' (Sanchez and Collins 1997) and/or of design protocols in CAD systems that constrain the ways in which components may be interrelated by product designers (Sanchez 1996c). These interface specifications and protocols represent 'design rules' that within some firms explicitly managing their knowledge resources are often supported by archival documents that give detailed statements of an organisation's knowledge about how components of different types behave and interact in a product architecture.

Creating an interface specification or design protocol that allows the substitution of component variations into a product architecture requires a high level of understanding about component interactions, i.e., substantial knowledge about component behaviours and their interactions at the architectural level. Thus, creating complete interface specifications forces a firm to document the state of a firm's technical understanding about the components that make up its products and their interactions – in effect creating a 'balance sheet' of the firm's knowledge resources that can be used in creating comparable kinds of products. Thus, just as lowering the quantities of parts in the production flow through a factory brings into clear view the bottlenecks and limitations in various activities of a production process, trying to create fully specified component interfaces in a modular product architecture reveals any 'bottlenecks' or limitations in an organisation's knowledge of how the components in its products behave and interact. Thus, creating modular architectures becomes a vehicle not only for clearly defining what a firm knows about components and their interactions (i.e., its component and architectural level knowledge as defined in its interface documents or design protocols), but also for identifying specific forms of new architectural knowledge which would improve the ability of the firm to create more flexible and robust modular architectures.

The loose coupling of component designs in a modular product architecture also improves the ability of component developers to learn about the behaviours of individual components through developing alternative component designs. By creating a stable information structure of standardised

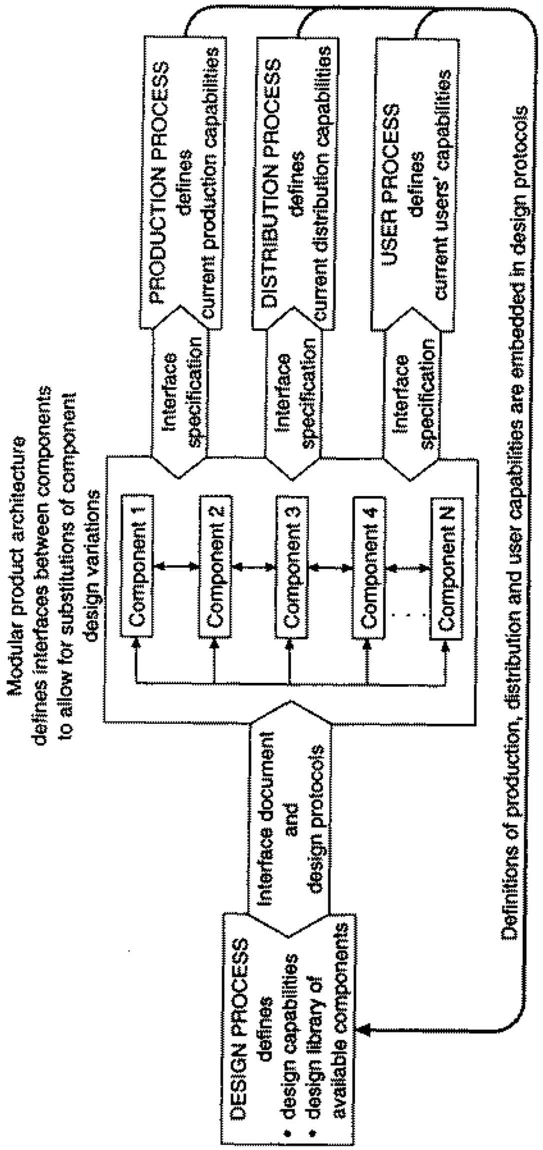

Figure 6.1 Modular product architecture as coordinating mechanism for modular process architectures

specifications governing input and output relationships between components within a modular product architecture, the complexity and uncertainty of the design context for each component may be considerably reduced (Sanchez and Tarondeau 1997), which in turn reduces the difficulty of learning about cause-and-effect relationships within each type of component. Thus, the loose coupling of component designs in a modular product architecture facilitates the discovery of less context-specific, more generalisable knowledge about individual components, which in turn leads to improved ability to identify and interrelate an organisation's component-level knowledge in creating architectural-level knowledge about how components will interact in a product architecture.

Support for the proposition that loose coupling of components within modular architectures enhances component-level and architectural-level learning is now appearing in a variety of forms. The modular architecture of the Boeing 777, for example, facilitated involvement of key airlines as partners in improving the component-level designs of cabin reconfiguration systems, engine servicing systems, baggage and freight loading systems, and other components whose development processes had been largely decoupled from development processes for other major components. The adoption by GE Fanuc Automation of modular open-systems architectures for its industrial controllers has enabled hundreds of component developers around the world to carry out loosely coupled, concurrent development processes creating improved components that 'plug and play' directly in GE Fanuc product architectures (Sanchez and Collins 1997).

Moreover, studies by Christensen and Bower (1996) and Christensen (1997) show that firms developing hard disk drives optimised for the specific requirements of large customers have generally failed to create the next generation of improved disk drives, which have typically been developed by firms creating disk drives for modular 'open architecture' computers. One explanation for these findings that is consistent with the propositions made here is that firms developing disk drives optimised to meet the specific requirements of a large customer's product architecture are creating relatively tightly coupled components with a more complex, tightly coupled learning environment that makes it more difficult to conceptualise, develop and test new component designs based on significantly different component technologies. Firms developing disk drives for a modular industry architecture, on the other hand, have a well-defined and bounded set of input and output specifications within which to learn about possible new component technologies and designs that would lead to 'generational' levels of improvement in disk drive performance.

For comparable reasons, adoption of a modular process architecture may enhance learning about loosely coupled activities within an organisation and about the desirable modes of interaction between loosely coupled units of activity. As the model in Figure 6.1 suggests, a modular product architecture provides a coordinating mechanism for achieving the loose coupling of an organisation's design, production, distribution and customer support activities.

As suggested in Figure 6.1, the design processes that lead to the creation of a product architecture are defined in terms of the specific design capabilities of the organisation. For example, an organisation may create a 'design library' (Sanchez and Mahoney 1996) of well-understood component designs with defined behaviours that can be used with confidence in a product architecture; the organisation may also define the design procedures it is confident can be used to create new component designs that can then be added to the design library.

Further, the capabilities of the organisation's available production processes may be defined and incorporated into design rules that clarify the freedom designers have to create new component variations that the organisation can readily produce. Similarly, the limitations which an organisation's distribution capabilities impose on the size, weight and other characteristics of a new product may be defined and incorporated into design rules. Current customer or user capabilities may also be defined and represented in the design rules governing the creation of new components and new product architectures.

The explicit definition of an organisation's capabilities in various process activities and of the resulting kinds of interactions which can feasibly be undertaken between those activities is directly analogous to defining component behaviours and their permissible modes of interaction in a modular product design. Once an organisation's process capabilities and their feasible modes of interaction are articulated in this manner,[4] the individual activities within an organisation become loosely coupled in the sense that decisions and changes made within one area of activity do not affect other areas of activity, so long as those decisions and changes stay within the range of input and output capabilities defined for that activity by the process architecture. Thus, within the limits of acceptable change defined by an organisation's modular process architecture, learning about processes within the knowledge domains of its loosely coupled activities becomes considerably less complex than trying to learn about those activities when they are tightly coupled (often in unspecified ways) to other activities in the organisation.

Just as a modular product architecture allows the quick leveraging of potentially many product variations by 'mixing and matching' new component variations within the product architecture, a modular process architecture enables the mixing and matching of existing process capabilities to reconfigure variations in organisational activity. Similarly, the explicit articulation of an organisation's process capabilities creates a frame of reference for identifying new process capabilities that would improve an organisation's *systemic* ability to respond to and take advantage of new technologies or market opportunities. Thus, creating an explicit modular process architecture helps an organisation to perceive and more precisely define opportunities for strategically important learning.

The next section discusses how creating modular product and process architectures creates a *modular knowledge architecture* that facilitates organisational learning.

4 Organisational learning within modular knowledge architectures

The preceding discussion has tried to explain how the creation of modular product architectures enables the loose coupling of modular processes for developing, producing and supporting products, how the creation of modular process architectures leads to a greater level of clarity in identifying and defining an organisation's capabilities and their interactions, and how this process of clarifying organisational capabilities facilitates organisational perceptions of opportunities for strategically useful learning that will improve its modular product and process architectures. This section introduces the concept of a *modular knowledge architecture,* which is defined here as *a knowledge structure composed of loosely coupled knowledge domains.* The discussion below considers the ways in which the modular knowledge architectures that result from adoption of modular product and process architectures improve processes for creating specific kinds of knowledge resources in an organisation.

Management researchers are now beginning to develop new concepts for representing different kinds of organisational knowledge (Grant 1993; Sanchez 1996a, 1997; Wright 1997). The following discussion elaborates on the notion that organisational knowledge may usefully be represented by categories of *know-how*, *know-why* and *know-what* (Sanchez 1996a, 1997) by showing how modular knowledge architectures may facilitate learning in these three categories of knowledge.

Table 6.2 provides a summary of know-how, know-why and know-what forms of organisational knowledge. A firm's know-how is its 'practical understanding' about the *current state of a system* (Simon 1981), where the system the organisation understands refers to its product or process architectures. Know-how knowledge enables a firm to continue operating its current systems, like leveraging product variations from its existing product architecture or running its production line within its existing process architecture.

Table 6.2 Know-how, know-why and know-what forms of knowledge

Form of knowledge	*Level of understanding*	*Capability derived from knowledge*
Know-how	'Practical understanding' of how a current system works	Enables firm to maintain operations within current product and process architectures
Know-why	'Theoretical understanding' of why a system works	Enables adaptation of current architectures or development of new architectures
Know-what	'Strategic understanding' of purposes to which know-why and know-how may be applied	Enables firm to imagine feasible new kinds of product and process architectures

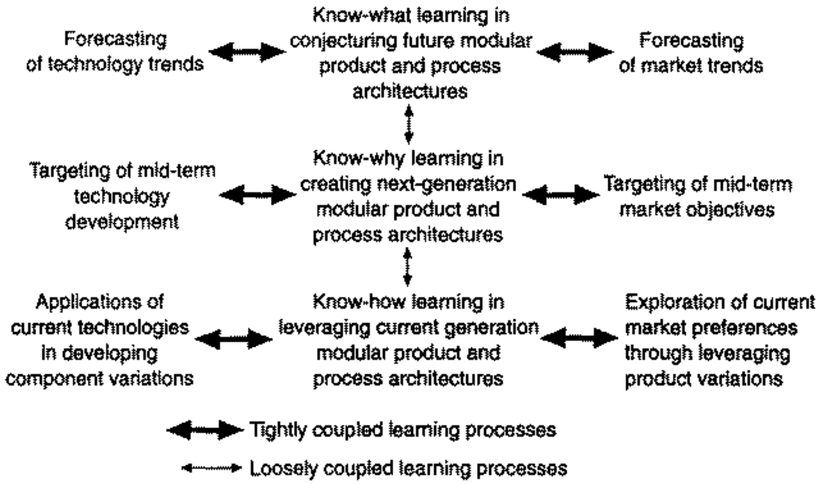

Figure 6.2 Loosely coupled learning processes for developing know-how, know-why and know-what knowledge

Know-how knowledge is useful in maintaining control of current product designs and production systems over some limited range of variations in internal or external conditions.

Know-why is 'theoretical understanding' of the *principles governing the functioning of a system* (Simon 1981). Know-why enables an organisation to change an existing system or to create a new system. Within product architectures as systems, know-why knowledge is the theoretical understanding of why specific components in a product architecture interact as they do. Similarly, within a process architecture as a system, know-why is understanding of the principles that determine how processes may be configured to work together. Know-why knowledge enables a firm to make significant changes in existing product or process architectures or to develop new product or process architectures. To create modular product or process architectures that permit substitution of many component variations requires well-developed know-why levels of knowledge about components or processes and their interactions.

Know-what is 'strategic understanding' of the *purposes* to which specific forms of know-why and know-how knowledge might be applied in creating product and process architectures (Sanchez 1996a, 1997). Know-what enables managers to make conjectures about what new kinds of products and processes a firm might develop and what the likely competitive impacts of specific new products or process capabilities might be. In essence, know-what knowledge is what enables managers to imagine and define feasible new kinds of products and processes for developing, producing and marketing products.

Figure 6.2 suggests ways in which modular knowledge architectures facilitate know-how, know-why and know-what forms of organisational or industry-

level learning processes. Within the modular knowledge architectures derived from modular product and process architectures, *know-how learning* may take place within loosely coupled component and process knowledge domains. Learning how to develop better components and better process capabilities within current product and process architectures is made easier and more efficient (Sanchez and Mahoney 1996) by intentionally decoupling know-how level learning from the next higher level of know-why learning. By using know-how level learning within current product and process architectures, organisations may lower the cost of exploring near-term market preferences by rapidly and cheaply introducing new product variations to test market reactions (Baldwin and Clark 1994; Sanchez and Sudharshan 1993) and may thereby accelerate learning about alternative component technologies through low-cost experimentation with product variations (Sanchez 1991, 1996b).

By using modular architecture to reduce the complexities and constraints that arise when introducing new component technologies or product market objectives into current product and process architectures, *know-why learning* can be focused on identifying and creating new knowledge about technologies and markets that will be useful in defining and developing the next generation of modular product and process architectures. Know-why learning can be channelled into efforts to understand the range of new technology and market possibilities that may be mediated most advantageously through alternative possibilities for next-generation modular product and process architectures of an organisation or an industry.

Similarly, *know-what learning* can be focused on long-term learning about technology and market trends. Long-term know-what learning occurs as a firm (or firms in an industry) tries, to define future modular product and process architectures that might mediate technology and market trends most advantageously. Although many organisations try to forecast technology and market trends, relatively few use those forecasts as inputs for conjecturing future product and process architectures. However, such a conjectural exercise may be a very effective way to give content and focus to the 'corporate imagination' (Hamel 1989b) and to defining specific targets for 'stretch and leverage' of organisational capabilities (Prahalad and Hamel 1993).

The basic proposition motivating Figure 6.2 and its attendant discussion is that learning at all three levels of know-how, know-why and know-what may be facilitated by the reduction of complexity at each level of knowledge, and such a reduction in complexity is made possible by the loose coupling of knowledge domains within modular product and process architectures. Recent empirical support for the learning benefits of creating loosely coupled knowledge domains within modular knowledge architectures is offered by two case studies of organisations using modular product and process architectures in knowledge-intensive businesses.

Post (1997) has studied the use of modular software architectures by Baan Company, a rapidly growing global software firm, and concludes that modularisation of Baan's software products not only makes possible a flexible modular organisation structure, but also improves Baan's ability to define and create new kinds of software products (know-what learning), to create the new kinds of architectures for those new kinds of products (know-why learning), and rapidly to develop new application program modules that exploit current software architectures to greatest effect (know-how learning).

Lang (1997) analysed the product development strategy of ARM Ltd, a small microprocessor design firm that nevertheless maintains a strong position in a global network of large semiconductor firms. ARM Ltd carries out successfully a strategy that focuses on developing and applying know-what and know-why knowledge in defining and developing modular microprocessor designs, while licensing its designs to large semiconductor firms with know-how knowledge about semiconductor production. Thus, the loose coupling of knowledge domains within the semiconductor industry makes it possible for some firms to focus on building certain forms of knowledge (like design know-why and market know-what) while others build up other kinds of knowledge (like production know-how).

5 Conclusions

Modular product and process architectures are increasingly prevalent in rapidly developing areas of technology and markets (Sanchez and Mahoney 1996). One explanation for this observed association may be that the adoption of modular product and process architectures within an industry creates modular knowledge architectures with substantial *positive network externalities* in organisational learning. In effect, use of modular knowledge architectures may create for all firms in an industry some efficiencies in building and leveraging knowledge. As firms in an industry join together to create industry standard specifications for components interactions, they thereby create a loose coupling of component designs that in turn creates a loose coupling of knowledge domains about components. A loose coupling of component knowledge domains can lead to faster, more efficient learning processes by firms focused on developing new components within an industry. Thus, adopting component-level standards within an industry may be a critically important mechanism for improving firm-level learning processes that, in the aggregate, may significantly accelerate the development of the industry's technological regime (Nelson and Winter 1982).

One observation in support of the proposition that adoption of industry standards for components and processes accelerates industry learning processes is that product markets with rapidly developing technologies often have well-established *de facto* or industry-sponsored standards for at least key component interfaces and often even for all the component interfaces in the industry's product architectures. Examples include the extensive hierarchy of interconnectivity standards in the telecommunications and personal computer industries and the use of Microsoft's Windows as the *de facto* standard architecture for personal computer applications software.

Modular product, process and knowledge architectures are 'industry resources' that are often 'taken for granted' by practitioners in rapidly evolving product markets and are rarely addressed directly by management researchers in studies of technological learning. Perhaps the role of modular architectures in creating knowledge infrastructures that accelerate organisational learning processes has yet to attract attention from management researchers that is commensurate with the impacts of modular architectures on technological learning in both firms and industries.

Notes

1 In more complex products, a product design may first be decomposed into a set of *subsystems* which are then decomposed into individual functional components (which are then further decomposed into subassemblies of parts and then into individual parts). Automobile companies, for example, typically decompose new product designs into a number of subsystems (like a passenger compartment climate control subsystem), each of which is further decomposed into major functional components (like a thermostatic control, an air conditioning compressor and a heater). Since the process of decomposition into subsystems typically involves a further decomposition into individual functional components, for simplicity I refer only to decomposition of product designs into components in this discussion.

2 *Hierarchical decomposability*, as used in this chapter, is essentially a *structural* concept. In this use, *hierarchy* has a broader meaning than that usually imputed in the organisational economics and strategic management literature (e.g., Williamson 1975; Mahoney 1992; Casson 1994), where hierarchies are characterised by an authority relation that governs the system. In the modular systems discussed in this paper, hierarchical decomposition and coordination are accomplished through the *information structure* (Sanchez and Mahoney 1996) of a product architecture rather than through the explicit use of an authority hierarchy as a coordinating mechanism.

3 Simon (1981) makes a similar distinction by referring to interdependent vs. 'semi-independent' elements of a system.

4 Like the articulation of component and architectural knowledge in a modular product architecture, articulation of an organisation's process architecture may take the form of process interface documents or computer-based process design protocols.

References

Baldwin, Carliss and Kim B. Clark (1994) 'Modularity-in-Design: an Analysis Based on the Theory of Real Options', Harvard Business School Working Paper, Boston, MA 02169, USA.

Bogner, William C. and Howard Thomas (1996) 'From Skills to Competences: The "Play-Out" of Resource-Bundles across Firms', in R. Sanchez, A. Heene and H. Thomas (eds), *Dynamics of Competence-Based Competition*, Oxford: Elsevier Pergamon.

Casson, Mark (1994) 'Why Are Firms Hierarchical?', *Journal of the Economics of Business* 1(1): 47–76.

Christensen, Clayton (1997) 'Patterns in the Evolution of Product Competition', *European Management Journal* 15(2): 117–27.

Christensen, Clayton and Joseph L. Bower (1996) 'Customer Power, Strategic Investment, and the Failure of Leading Firms', *Strategic Management Journal* 17: 197–218.

Dosi, Giovanni (1982) 'Technological Paradigms and Technological Trajectories', *Research Policy* 11: 147–62.

Farrell, J. and G. Saloner (1985) 'Standardization, Compatibility, and Innovation', *Rand Journal of Economics* 16: 70–83.

Foss, Nicolai and Bo Eriksen (1995) 'Competitive Advantage and Industry Capabilities', in Cynthia Montgomery (ed.), *Resource-Based and Evolutionary Theories of the Firm: Towards a Synthesis*, Boston, MA: Kluwer Academic.

Garud, Raghu and Arun Kumaraswamy (1993) 'Changing Competitive Dynamics in Network Industries: An Exploration of Sun Microsystems' Open Systems Strategy', *Strategic Management Journal* 14: 351–69.

Grant, Robert M. (1993) 'Organizational Capability within a Knowledge-Based View of the Firm', School of Business Administration Working Paper Series, STRAT-2277-03-1293, Georgetown University, Washington, DC 20057.

Hamel, Gary (1989a) 'Strategic Intent', *Harvard Business Review* 67: 63–76.

—— (1989b) 'Corporate Imagination and Expeditionary Marketing', *Harvard Business Review* 69: 81–92.

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

Lang, John W. (1997) 'Leveraging Knowledge across Firm Boundaries: Achieving Strategic Flexibility through Modularization and Alliances', in R. Sanchez and A. Heene (eds), *Strategic Learning and Knowledge Management*, Chichester: John Wiley and Sons.

Langlois, Richard N. and Paul L. Robertson (1992) 'Networks and Innovation in a Modular System: Lessons from Microcomputer and Stereo Components Industries', *Research Policy* 21: 297–313.

Lee, H. L., C. Billington and B. Carter (1993) 'Hewlett-Packard Gains Control of Inventory and Service through Design for Localization', *Interfaces* 23: 1–11.

Mahoney, Joseph T. (1992) 'Organizational Economics within the Conversation of Strategic Management', *Advances in Strategic Management* 8: 103–55.

Malone, Thomas W., Joanne Yates and Robert Benjamin (1987) 'Electronic Markets and Electronic Hierarchies', *Communications of the ACM* 30(6): 484–97.

Meyer, Marc H. and James Utterback (1993) 'The Product Family and the Dynamics of Core Capability', *Sloan Management Review* 34(3): 29–47.

Nelson, Richard and Sidney Winter (1982) *An Evolutionary Theory of Economic Change*, Cambridge, MA: The Belknap Press.

Orton, J. Douglas and Karl E. Weick (1990) 'Loosely Coupled Systems: A Reconceptualization', *Academy of Management Review* 15(2): 203–23.

Post, Henk A. (1997) 'Modularity in Product Design, Development, and Organization: A Case Study of Baan Company', in R. Sanchez and A. Heene (eds), *Strategic Learning and Knowledge Management*, Chichester: John Wiley and Sons.

Prahalad, C. K. and Gary Hamel (1993) 'Strategy as Stretch and Leverage', *Harvard Business Review* March–April: 75–84.

Sanchez, Ron (1991) 'Strategic Flexibility, Real Options, and Product-Based Strategy', Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA 02139.

—— (1994) 'Towards a Science of Strategic Product Design: System Design, Component Modularity, and Product Leveraging Strategies', *Proceedings of the Second International Product Development Management Conference on New Approaches to Development and Engineering*, May 30–31, 1994, European Institute for Advanced Studies in Management, Brussels, Belgium.

—— (1995) 'Strategic Flexibility in Product Competition', *Strategic Management Journal* 16, Summer Special Issue: 135–59.

—— (1996a) 'Strategic Product Creation: Managing New Interactions of Technologies, Markets, and Organizations', *European Management Journal* 14(2): 121–38.

—— (1996b) 'Integrating Technology Strategy and Marketing Strategy', in H. Thomas and D. O'Neal (eds), *Strategic Integration*, Chichester: John Wiley and Sons.

—— (1996c) 'Quick-Connect Technologies for Product Creation: Implications for Competence-Based Competition', in R. Sanchez, A. Heene and H. Thomas (eds), *Dynamics of Competence-Based Competition*, Oxford: Elsevier Pergamon.

—— (1997) 'Managing Articulated Knowledge in Competence-Based Competition', in R. Sanchez and A. Heene (eds), *Strategic Learning and Knowledge Management*, Chichester: John Wiley and Sons.

Sanchez, Ron and Robert P. Collins (1997) 'Competing in Modular Markets', Working Paper, Department of Industrial Economics and Strategy, Copenhagen Business School, DK-1366 Copenhagen, Denmark.

Sanchez, Ron and Aimé Heene (1996a) 'A Systems View of the Firm in Competence-Based Competition', in R. Sanchez, A. Heene and H. Thomas (eds), *Dynamics of Competence-Based Competition*, Oxford: Elsevier Pergamon.

—— (1996b) 'Competence-Based Strategic Management: Concepts and Issues for Theory, Research, and Practice', in A. Heene and R. Sanchez (eds), *Competence-Based Strategic Management*, Chichester: John Wiley and Sons.

Sanchez, Ron, Aimé Heene and Howard Thomas (1996) 'Towards the Theory and Practice of Competence-Based Competition', in R. Sanchez, A. Heene and H. Thomas (eds), *Dynamics of Competence-Based Competition*, Oxford: Elsevier Pergamon.

Sanchez, Ron and Joseph T. Mahoney (1994) 'The Modularity Principle in Product and Organization Design', Office of Research Working Paper no. 94-0157, University of Illinois, Champaign, IL 61820.

—— (1996) 'Modularity, Flexibility, and Knowledge Management in Product and Organization Design', *Strategic Management Journal* 17, Winter Special Issue: 63–76.

Sanchez, Ron and D. Sudharshan (1993) 'Real-Time Market Research: Learning-by-Doing in the Development of New Products', *Marketing Intelligence and Planning* 11, August: 29–38.

Sanchez, Ron and Jean-Claude Tarondeau (1997) 'Modular Product Design and the Management of Uncertainty in Innovation and Product Development', Working Paper, Graduate School of Management, University of Western Australia, Nedlands, WA 6907.

Sanderson, Susan Walsh and Mustafa Uzumeri (1997) *Managing Product Families*, Chicago: Irwin.

Saxonian, A. (1991) 'The Origins and Dynamics of Production Networks in Silicon Valley', *Research Policy* 20: 423–37.

Simon, Herbert (1981) *The Sciences of the Artificial*, Cambridge, MA: MIT Press.

Teece, David (1992) 'Foreign Investment and Technological Development in Silicon Valley', *California Management Review* 34: 88–106.
von Hippel, Eric (1994) 'Sticky Information and the Locus of Problem Solving', *Management Science* 40: 429–39.
Wheelwright, Steven C. and Earl Sasser, Jr. (1989) 'The New Product Development Map', *Harvard Business Review* May–June: 112–22.
Williamson, Oliver E. (1975) *Markets and Hierarchies: Analysis and Anti-trust Implications*, New York: The Free Press.
Wright, Russell W. (1997) 'Tangible Integration versus Intellectual Codification Skills: A Comparison of Learning Processes in Developing Logic and Memory Semiconductors', in R. Sanchez and A. Heene (eds), *Strategic Learning and Knowledge Management*, Chichester: John Wiley and Sons.

Resources, Technology and Strategy

Explorations in the resource-based perspective

Edited by Nicolai J. Foss and Paul L. Robertson

London and New York

First published 2000
by Routledge
11 New Fetter Lane, London EC4P 4EE

Simultaneously published in the USA and Canada
by Routledge
29 West 35th Street, New York, NY 10001

Routledge is an imprint of the Taylor & Francis Group

Typeset in Times by
The Running Head Limited, www.therunninghead.com
Printed and bound in Great Britain by
MPG Books Ltd, Bodmin

British Library Cataloguing in Publication Data
A catalogue record for this book is available
from the British Library

Library of Congress Cataloging in Publication Data
Resources, technology, and strategy/[edited by] Nicolai J. Foss and
Paul L. Robertson
(Routledge advances in management and business studies)
Includes bibliographical references and index.
1. Industrial management. 2. Strategic planning
3. Technological innovations – Management. 4. Human capital
I. Foss, Nicolai J., 1964– . II. Robertson, Paul L. III. Series
HD31.R464 2000
658—dc21 99-32605 CIP

ISBN 0-415-21585-4