13

Modular Product and Process Architectures

Frameworks for Strategic Organizational Learning

Ron Sanchez

In recent years management theory and practice have begun to take note of the increasingly important role of product architectures in product market competition (Morris and Ferguson 1993, Sanchez 1995, Sanderson and Uzumeri 1997). As growing numbers of firms adopt architectural approaches to managing their products and processes, however, some managers and researchers have begun to understand that architectural approaches to creating new products and processes can offer a substantial benefit beyond new product strategies and improved market success. Architectures can also help organizations to identify and manage key knowledge assets and to discover their most strategically beneficial opportunities for acquiring new knowledge and capabilities. Among the architectures that a firm might adopt, *modular product and process architectures* offer an especially powerful framework both for managing current knowledge assets effectively and for guiding strategic organizational learning (Sanchez and Mahoney 1996). This chapter investigates the nature of modular product and process architectures and their roles in supporting effective knowledge management and strategic organizational learning. Our discussion is organized in the following way.

A presumption often evident in much writing on knowledge management is that firms may lose competitive advantage by converting their tacit knowledge into articulated, explicit knowledge. I therefore first consider some fundamental arguments why organizations should try to make key forms of knowledge explicit and organizational rather than leaving such knowledge in tacit, personal form. This discussion provides the essential strategic justifications for using architectures to identify and articulate an organization's strategically important knowledge used in creating new products and processes.

I next define product and process architectures and identify the key forms of technical and market knowledge used in creating product and process architectures. I then consider the special characteristics of *modular* architectures and how processes for creating and using modular architectures help an organization to develop its tech-

nical and market knowledge and make those key forms of knowledge explicit within the organization.

I then explain how using modular architectures to make knowledge explicit can also help an organization to discover the knowledge deficits and "capability bottlenecks" that limit its ability to create and realize new products. Extending this architectural perspective on organizational learning, I identify four forms of strategic learning that an organization can undertake to overcome its current capability bottlenecks.

This chapter concludes by explaining the essential features of a new approach to managing product creation processes that uses modular architectures systematically to identify, leverage, and continuously improve an organization's technical and market knowledge. (Examples of modular architectures at work may be found in Sanchez [2001] and Sanchez [1999].)

Organizational Knowledge: Better Left Tacit, or Better Made Explicit?

It is often asserted by strategic management scholars writing on knowledge management that an organization's strategically important knowledge is best left in tacit, personal form. The basic argument in support of this approach to managing knowledge goes like this: if strategically important knowledge held by individuals in an organization is articulated and made explicit, that knowledge can readily diffuse outside the organization, be absorbed by competitors, and lose its ability to be a source of competitive advantage once all competitors possess and use the knowledge.

While seemingly plausible, observation of knowledge management practices in many firms suggests that this argument greatly overestimates the potential for knowledge that is articulated and made explicit in one organization to be absorbed and used by competing organizations. In addition, this argument ignores the significant risks and competitive disadvantages of leaving knowledge that is important to an organization in tacit, personal form (Sanchez 1997). As a preface to the discussion of the role of architectures in helping organizations to make important forms of knowledge explicit, I summarize below the arguments that support knowledge management strategies premised on making strategically important knowledge explicit within an organization.

First consider the inherent limitations and risks to an organization of leaving important knowledge in tacit form. When key knowledge—such as technical and market knowledge about an organization's products—is left in tacit form in the minds of individuals, the ability of the organization to use that knowledge advantageously is limited by the amount of time each individual can spend in personally applying his or her knowledge in the value-creation processes of the organization. Moreover, applying the personal knowledge of individuals in new projects or trying to transfer one employee's tacit knowledge to other employees generally requires moving key "knowledge workers" from one location to another—something that is generally costly and time consuming and may meet with resistance from those employees. More critically, an organization will lose knowledge that is left in tacit form when a person having such knowledge leaves the organization. In essence, leaving strategically important knowledge in tacit form exposes an organization to the risk that critical knowledge will simply "walk out the door" as employees retire, fall ill, change jobs, or—in the worst case scenarios—take positions in competing firms. Even if a person with important tacit knowledge remains in an organization, he or she may use the threat of withholding that knowledge to acquire organizational power and advance his or her personal interests at the expense of the organization as a whole (Salancik and Pfeffer 1977).

By contrast, articulating the tacit knowledge of individuals into explicit organizational knowledge makes it possible to disseminate knowledge rapidly and widely within the organization. As an intellectual asset, explicit knowledge is free of the limitations of time and space that constrain the use of physical assets (including human assets with tacit knowledge). Once articulated, explicit knowledge may be shared with many employees quickly and globally through information technology and communication systems. Compared to relying on moving individual "tacit knowledge carriers" around an organization, electronically disseminating explicit knowledge throughout an organization enables strategically important knowledge to be applied extensively throughout an organization and with maximum speed. Making key forms of knowledge explicit also helps all members of an organization to discuss, debate, test, and improve the organization's knowledge—something that cannot be done when knowledge remains in tacit form in the

mind of an individual. Finally, articulating the tacit knowledge of individuals into the explicit knowledge of an organization can greatly mitigate the potential for individuals to use their personal tacit knowledge to acquire and use power for personal advantage within an organization.

In principle, explicit knowledge that diffuses beyond the boundaries of an organization and is absorbed and used by competitors could lose its value as a source of competitive advantage. In practice, however, there are a number of reasons why knowledge that has been made explicit within an organization would be unlikely to lose its strategic value in this way. First, because organizations often create their own contextual "corporate languages" for discussing key issues and strategically important factors (von Krogh et al. 1994), knowledge may be articulated in terms that are readily understood within an organization but not understood by people outside the organization. For example, I once worked for a trade association that had a set of 13 rules governing the interactions of employees with people outside the association. Twelve of the rules provided guidelines for dealing with specific kinds of situations, but the rule 13 literally said, "If none of the 12 prior rules seems to apply, do something intelligent." Over time, stories of how employees had successfully followed "rule 13" in various situations created a legacy of examples that gave specificity to the notion of "doing something intelligent" in similar situations. In this way, the phrase "rule 13" stood for a set of actions that were explicitly understood internally by members of the organization, but the phrase conveyed no meaning to people outside the organization.

Even when knowledge is articulated in the terms of a standard language that is not unique to a given organization, the full meaning of explicit statements of some form of knowledge may not be grasped by people with experience, education, and capabilities different from those of the people who articulated the knowledge. Chemical formulas, engineering drawings, design methodologies, and other articulated forms of knowledge used within a firm may not be understood by outsiders—even competitors—with different skill sets, knowledge bases, and work processes. Moreover, even when the meaning of an explicit statement of knowledge is understood by someone, it is not at all certain that this person will recognize the potential strategic value of that knowledge when applied in the most effective way in its best uses. Further, simply understanding something intellectually is not sufficient to enable effective application of the knowledge in practice (Heene 1993). Applying some explicit knowledge effectively may require a complex set of supporting skills and assets that even the direct competitors of an organization may lack. In effect, because the value of any form of knowledge derives from its use in a system of complementary skills and assets, a bit of explicit knowledge that may be of great strategic value in one organizational context may be of limited usefulness and value in another organizational context (Sanchez and Heene 1997). Finally, if appropriate steps are taken to control important forms of explicit knowledge within an organization—such as Intel's policy of providing employees access to explicit knowledge only on a "need-to-know" basis—the risk that explicit knowledge will diffuse to competitors in the first place can be greatly moderated.

Boisot (1995) has coined the term "paradox of value" to refer to the simultaneous benefits and risks that an organization may face when it makes its knowledge explicit. The analysis here suggests, however, that the risks of making knowledge explicit within an organization may be less threatening and more manageable than commonly imagined. The analysis further suggests that with proper management of an organization's explicit knowledge, the "paradox of value" may be transformed into something more like the biblical "miracle of the loaves," in which one person's knowledge, once made explicit, may be shared among and help to sustain thousands of people within an organization.

With these perspectives on the strategic limitations and risks of *not* making organizational knowledge explicit, as well as the strategic benefits that may be obtained when knowledge is made explicit within an organization, the next section considers ways in which well-defined product and process architectures can help organizations to identify and articulate key organizational knowledge used in creating new products and processes.

Product and Process Architectures in Knowledge Articulation

The term "architecture" refers to two fundamental properties of a product or process design. (1) The way in which a product design is decomposed into *functional components* and the way in which a process design is decomposed into

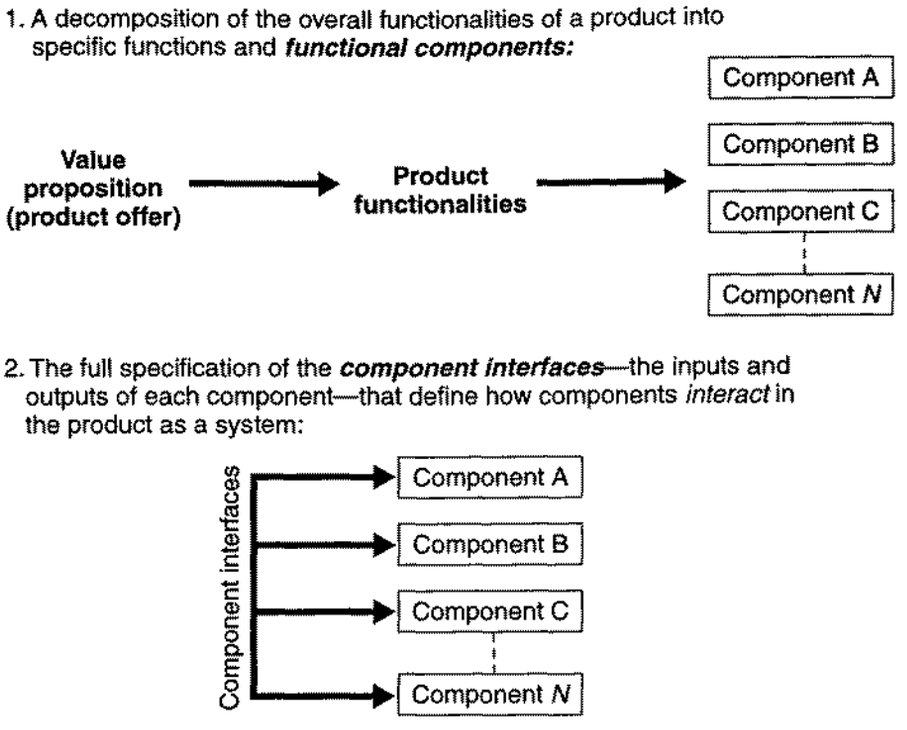

Figure 13.1 Two elements of a product architecture.

functional activities. For simplicity in this discussion, I will use the term "components" to refer to the functional elements of both product and process designs. (2) The ways in which the functional components interact in a product or process design. In the designs of products and processes, the interactions of functional components and activities are represented in the form of *interface specifications* that define how components or activities "interface" or interact with each other when they function together as a system[1] (Sanchez 1995, 1999, 2002). Figures 13.1 and 13.2 illustrate the decomposition of product and process designs into functional components and the use of interface specifications to define how components interact in those designs.

Carefully defining the architectures of its products and processes helps an organization to articulate strategically important knowledge in at least two ways. First, organizations create designs of products and processes in an effort to create "value propositions"—that is, product offers that create value for customers. By carefully analyzing how each kind of functional component in its product and process designs creates value for customers, an organization can develop deeper insights into ways that its technological capabilities can be applied to create specific customer benefits. Thus, analyzing how the overall benefits a firm seeks to bring to customers can best be decomposed into specific product and process components can lead to much clearer, more precise, and more explicit understanding of how the firm's technology can be applied to create value for targeted customers.

Of course, a collection of components must function together effectively and reliably as a system to create benefits and value in the marketplace. Creating effective, reliable products and processes therefore requires knowledge about the *system behaviors* of the components an organization uses in its products and processes. The system behaviors of components are defined and controlled through the interface specifications an organization establishes for its products and processes. In organizations that are focused on creating products rather than on understanding and managing the component structure of their product designs, interface specifications are usually regarded as a technical detail to be worked out during product development. In some organizations, the interface specifications for their various product designs may not even be clearly documented at the end of a development process. An organization that adopts an architectural approach to managing product creation, however, will soon understand that the interface specifications it develops for the components it uses constitute a "balance sheet" of the organization's explicit knowledge about the system behaviors

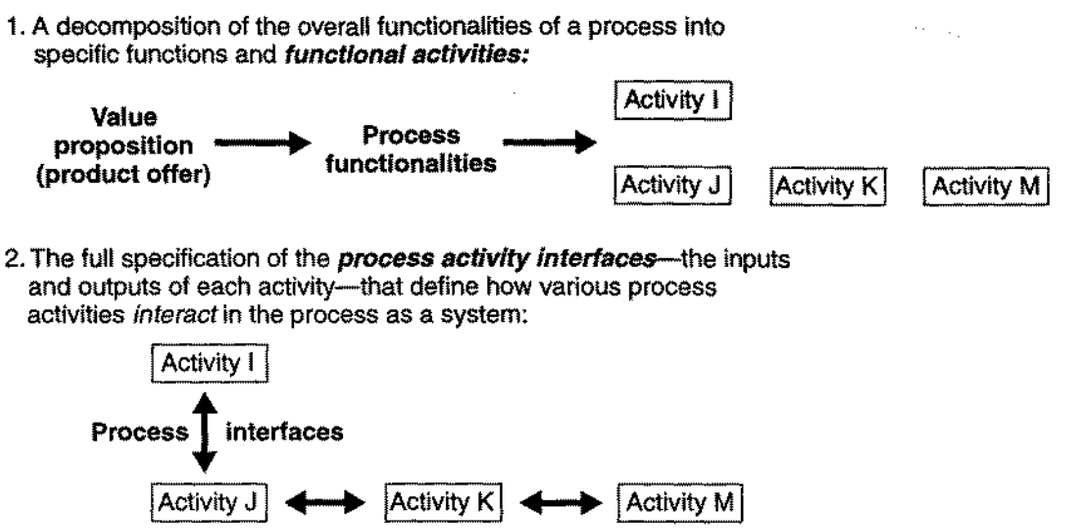

Figure 13.2 Two elements of a process architecture.

of those components when used in the organization's products and processes. Interface specifications therefore become a primary vehicle for articulating and making explicit the organization's collective technical knowledge about the system behaviors of the components it uses in its products and processes.

Modular Product and Process Architectures in Learning and Knowledge Articulation

Modular architectures are a special kind of architecture in which the interfaces between components have been specified to allow the *substitution of a range of component variations* into an architecture and thereby enable the ready configuration of a range of product variations based on different combinations of components (Langlois and Robertson 1992, Garud and Kumaraswamy 1993, Sanchez 1995). The possibility of "mixing and matching" a range of "plug-and-play" compatible components within a single modular architecture suggests new kinds of product strategies and new possibilities for product creation processes (Sanderson and Uzumeri 1997). Rather than focusing product strategy and development processes on creating individual new products, organizations may create modular architectures that can serve as "platforms" for leveraging a broad range of product variations based on combinations of different component variations.

The leveraging of product variations based on mix-and-match combinations of plug-and-play modular components makes possible new marketing strategies that support new forms of market learning (Sanchez 1999). Modular architectures may enable a firm to learn about consumer preferences in "real time" by testing consumer reactions to a range of modular product variations. Modular architectures may enable the leveraging of larger numbers of product variations to support more extensive market exploration and segmentation.

Modular architectures may also be designed to allow the substitution of higher performing components to support the rapid technological upgrading of products as higher performing components are developed in the future. In addition, modular architectures may be used to lower overall costs of maintaining high levels of product variety and change through careful use of common components in many product models and by reusing component designs in successive generations of products (Sanchez 1995, 1999, Sanchez and Sudharshan 1993).

Creating modular architectures to support these new kinds of product strategies can greatly stimulate and deepen an organization's technical learning about both component designs and interface specifications. When a firm follows a traditional product-focused development process, its primary concern during product development is to devise specific component designs that will enable each component to perform its function in the desired new product. In traditional product development, an organization's learning about components therefore tends to become narrowly focused on finding "local," context-dependent component design solutions that will

work in a specific product design. Such learning tends to produce "sticky" contextual knowledge about components that often cannot readily be transferred to other product development projects (von Hippel 1994).

By contrast, when an organization pursues a modular development process in which the output is conceived as an architecture that will serve as a platform for leveraging product variations based on mix-and-match combinations of component variations, the organization will have an incentive to explore the full set of component designs that may be used in the modular architecture. Creating modular architectures rather than single new product designs therefore tends to result in broader organizational learning about feasible component designs. This expanded component-level learning helps to develop broader and more explicit organizational knowledge about the feasible range of variations in functions, features, performance levels, and costs that various kinds of components can bring to the organization's products.

Modular architectures also stimulate more extensive and explicit organizational learning about the behaviors and interactions of components in products and process designs. When a firm follows a conventional product development process, interface specifications tend to be viewed in a narrowly pragmatic way. The primary (and sometimes only) concern in product-focused development processes is to specify a set of component interfaces that will enable a given set of component designs to work together reliably in a specific new product design. In this mode, learning about the systemic behaviors of the different types of components an organization uses in its products tends to become "spotty" and contextually dependent on the specific component designs used in the product designs a firm has developed.

However, when a firm creates modular architectures intended to support the plug-and-play mixing and matching of a broad range of component variations, there are incentives—and indeed imperatives—to develop more fundamental knowledge about how different kinds of component variations will behave when combined with other kinds of component variations. Learning about component interfaces therefore becomes elevated from solving a problem of controlling the interactions of specific components used in a single product design to developing broader, more explicit knowledge about the behaviors and systemic interactions of many kinds of components and component variations that is needed to define robust interface specifications for a modular architecture.

Because creating modular architectures requires more fundamental, more explicit knowledge of feasible component designs and the systemic behaviors of components than is required when an organization simply develops single new product designs, creating modular architectures helps an organization to discover the current limits of its fundamental knowledge about components and their interactions. These technical knowledge deficiencies implicitly act as "capability bottlenecks" that limit an organization's ability to create and realize new products and to launch more aggressive, exploratory, and flexible product strategies. By using modular development processes to define more explicitly what an organization currently knows about the components it may use in its product architectures, the organization can begin to "see" more clearly strategically useful forms of technical knowledge that it currently lacks.

Creating modular architectures can therefore bring an organization two important knowledge management benefits. Modular architecture development processes help an organization to define more explicitly "what we know" as an organization about creating new component designs and controlling their interactions within modular architectures. With a clearer understanding of what it is currently capable of in this regard, an organization can leverage its current knowledge more quickly, widely, and effectively in creating flexible, robust product and process architectures. At the same time, by using modular architecture development processes to make explicit the limits of "what we know about component behaviors" an organization can more readily identify specific opportunities for developing new technical knowledge in component design and interface specification that can overcome the organization's current capability bottlenecks and bring definable strategic benefits to the organization.

Four Forms of Strategic, Systematic Learning in Creating Modular Architectures

An architectural perspective on the knowledge used in creating new products and processes suggests that there are four forms of strategic learning that an organization can undertake in overcoming its current capability bottlenecks. As outlined in figure 13.3, organizational learning in an architectural framework may consist of

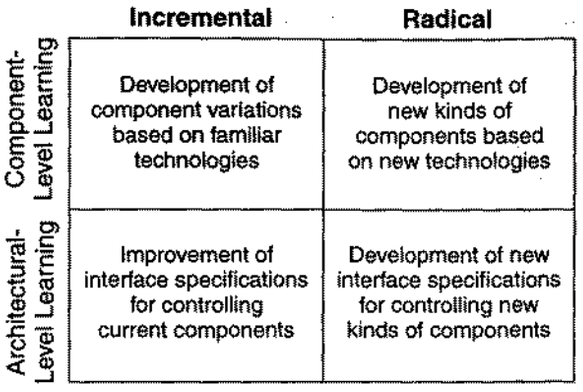

Figure 13.3 Four forms of learning in an architectural framework.

component-level learning and architectural-level learning. Component-level learning may take the form of either "incremental" learning about current kinds of components or more "radical" learning about new kinds of components. In general, incremental component-level learning occurs when an organization develops variations on existing component designs based on familiar technologies, while radical component-level learning occurs when developing new designs for new kinds of components that use new technologies to bring new kinds of functionalities to a product.

Similarly, incremental architectural-level learning involves the development of better interface specifications that improve and extend the ability of an organization to control the system behaviors of the kinds of components it currently uses in its products. Radical architectural-level learning occurs when an organization develops interface specifications that enable a new kind of component (usually based on new technology and/or providing new functionality) to be used reliably within a product architecture.

These four forms of technical learning may be managed strategically in a new kind of development process that systematically uses modular architectures as a framework for managing organizational learning, as summarized in figure 13.4. In this modular approach to strategically managing technical learning, the organizational processes for undertaking each form of component and architectural learning are carefully separated and coordinated (Sanchez and Mahoney 1996, Sanchez and Collins 2001). As illustrated in the lower "learning loop" of figure 13.4, incremental learning about components takes place during the leveraging of a current modular product architecture. As a firm uses the flexibility of its current modular architecture to explore customer reactions to different combinations of component-based functions, features, and performance levels offered at different price points, a firm may improve its knowledge of the component-based benefits consumers most desire from the firm's product and of the price elasticity of consumer demand for those benefits. This market knowledge may then drive incremental component-level learning through developing more component variations to serve the strongest, most profitable consumer preferences for component-based benefits.

In the top learning loop of figure 13.4, the long-term strategic planning processes of an organization integrate technology road mapping and market trend forecasting to define future generations of modular architectures that would try to take greatest advantage of future technological possibilities to serve future market opportunities. Defining future generations of modular architectures enables a firm to identify new kinds of components that could be based on new technologies expected to be available in the future and that would provide new kinds of functions and benefits to customers. The organization's long-term strategic learning can then be focused on radical learning about how to design the expected new kinds of components and how to specify interfaces that will control the system behavior of the new kinds of components when used with other components.

Unlike traditional research and development that produces only "proof of concept" and then relies on product development processes to create reliable designs for new components, these two forms of radical learning lead to a new form of organizational knowledge that may best be called *proof of component*—that is, the ability to specify and control the system behavior of a new kind of component. New kinds of components—once their basic designs are established and system behaviors understood—may then be placed in an organization's "design library" of proven, explicitly understood component designs that can be used in developing the next generations of product architectures.[2]

The middle section in figure 13.4 represents the incremental architectural learning involved in improving a firm's interface specifications for components to be used in an organization's next-generation product architecture and the incremental component learning involved in creating new component variations (but not new kinds of components) to be used in a next-generation product architecture. This incremental learning benefits from feedback about the reliability and

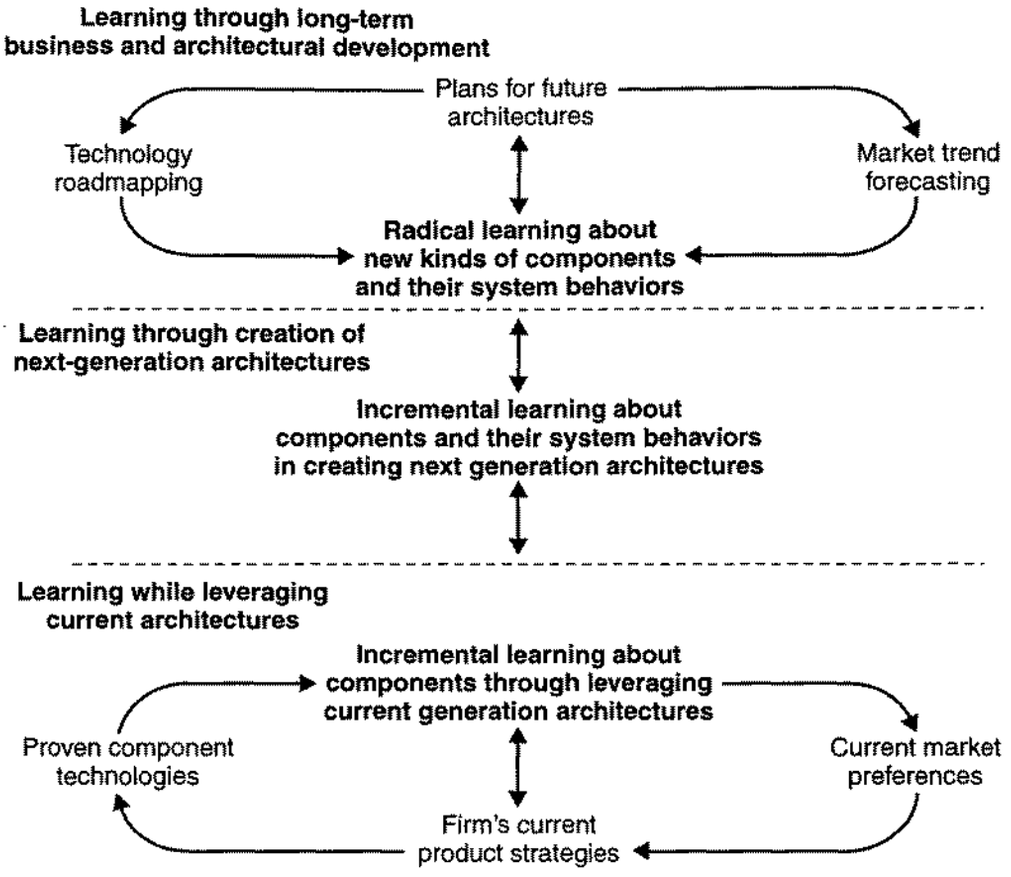

Figure 13.4 Architectures as frameworks for systematic, strategic learning.

performance of its current architecture gained through experiences in leveraging product variations from its current product architecture. Incremental learning in next-generation architecture development also benefits from "feed-forward" from its radical learning processes that have developed new component designs with proven system behaviors.

This new modular approach to managing product creation rests on the principle of making explicit the knowledge within an organization about the kinds of components that can produce benefits for customers, the kinds of component designs that can perform reliably in given kinds of product and process architectures, and the interface specifications needed to assure reliable performance and interactions of each component in a system of components.

Conclusion

This discussion has suggested several ways in which adopting a modular architectural approach to product development can stimulate organizational learning about markets and technologies. The need to understand and control components' system behaviors within a modular architecture also helps to make the collective technical knowledge of an organization more explicit and, by being explicit, more manageable strategically. Once made explicit, the current technical knowledge of an organization can be leveraged more effectively, quickly, and widely through the configuration of an expanded range of product variations based on mixed-and-matched, plug-and-play modular component variations. Once the current technical knowledge of an organization is made explicit, the limits of that knowledge and its associated capability bottlenecks can be recognized more clearly. Organizational learning may then be strategically focused on developing new knowledge that will remove those capability bottlenecks. In some firms, these strategic learning processes are now being organized in a new model of product creation that uses the creation of modular architectures as a framework for driving both incremental and radical learning about components and their interactions (Sanchez and Collins 2001).

Notes

1. Interface specifications define and manage the following seven kinds of interactions between product components: (1) how components physically *attach* to each other; (2) what *space* within the overall arrangement of components in a product a given component will occupy; (3) what is *transferred* into and out of a component as the component performs its primary function in the product as a system (e.g., transformation of electrical power, of chemical energy into mechanical energy, of a bit stream from one format to another, etc.); (4) how one component *signals* to a second component what state it is in, and how the second component signals to the first component whether to stay in that state or change to another state—this interface generally includes the critical *timing* coordination among components in a product with internal dynamics; (5) how each component interacts with the *user* of the product; (6) how each component must interact with the intended *ambient environment*—that is, the intended environmental context of use of the product (intended range of temperature, humidity, shock, vibration, and so forth; and (7) how each component will interact with the *internal environment* of the product in unintended ways—that is, how the functioning of each component may generate heat, emit electromagnetic fields, or produce effects that can affect the functioning of other components in undesirable ways. For a more detailed discussion of component interfaces, see Sanchez (2002).

2. Radical component and architectural learning is a form of technology development that some firms are now defining as an intermediate process between traditional research and development and traditional product development. Achieving "proof of component" before using components in product development essentially removes most technological uncertainty from product development per se (or more precisely, from next-generation architecture development), and as a result can greatly accelerate product development and make its outcomes much more predictable (Sanchez and Collins 2001, Sanchez 2002).

References

Boisot, M.H. (1995) *Information Space*. London: Routledge.

Garud, R. and Kumaraswamy, A. (1993) "Changing competitive dynamics in network industries: an exploration of Sun Microsystems' open systems strategy." *Strategic Management Journal* 14(5): 351–369.

Heene, A. (1993) "Classifications of competence and their impact on defining, measuring, and developing 'core competence'." Paper presented at the second International Workshop on Competence-Based Competition, EIASM, Brussels, November.

von Hippel, E. (1994) "Sticky information and the locus of problem solving." *Management Science* 40(4):429–439.

von Krogh, G., Roos, J., and Slocum, K. (1994) "An essay on corporate epistemology." *Strategic Management Journal* 15:53–71.

Langlois, R.N. and Robertson, P.L. (1992) "Networks and innovation in a modular system: lessons from the microcomputer and stereo component industries." *Research Policy* 21(4):297–313.

Morris, C.R. and Ferguson, C.H. (1993) "How architecture wins technology wars." *Harvard Business Review* 71(2):86–96.

Salancik, G.R. and Pfeffer, J. (1977) "Who gets power—and how they hold on to it: a strategic-contingency model of power." *Organizational Dynamics* (Winter):3–21.

Sanchez, R. (1995) "Strategic flexibility in product competition." *Strategic Management Journal* 16(Summer):135–159.

Sanchez, R. (1997) "Managing articulated knowledge in competence-based competition," in R. Sanchez and A. Heene (eds.), *Strategic Learning and Knowledge Management*, pp. 163–187. Chichester: Wiley.

Sanchez, R. (1999) "Modular architectures in the marketing process." *Journal of Marketing* 63:92–111.

Sanchez, R. (2001) *Modularity, Strategic Flexibility, and Knowledge Management*. Oxford: Oxford University Press.

Sanchez, R. and Collins, R.P. (2001) "Competing—and leaving—in modular markets." *Long-Range Planning*. in press.

Sanchez, R. and Heene, A., eds. (1997) *Strategic Learning and Knowledge Management*. Chichester: Wiley.

Sanchez, R. and Mahoney, J.T. (1996) "Modularity, flexibility, and knowledge management in product and organization design." *Strategic Management Journal* 17(Winter):63–76.

Sanchez, R. and Sudharshan, D. (1993) "Real-time market research: Learning-by-doing in the development of new products." *Marketing Intelligence and Planning* 11(7):29–38.

Sanderson, S.W. and Uzumeri, M. (1997) *Managing Product Families*. Chicago: Irwin.

The Strategic Management of Intellectual Capital and Organizational Knowledge

Edited by

Chun Wei Choo

Nick Bontis

OXFORD
UNIVERSITY PRESS

2002