10
Coupling the Technical and Institutional Faces of Janus in Network Industries

RAGHU GARUD
ARUN KUMARASWAMY

Technology pervades almost every facet of our daily lives, yet we seldom pause to reflect upon how a technology works, our cognitive abilities and interests circumscribing the complexity that we can or want to confront. Seldom, too, do we pause to inquire into the origins of a technology, its roots and subsequent evolution becoming opaque to scrutiny with the passage of time. As these facets become taken for granted, the entire technology becomes a black box (Rosenberg, 1982).

Occasionally we, as regulators, policymakers, and technology users, attempt to pry open the black box and ask: How do technologies evolve? At first, we might be inclined to adopt a functional perspective, assuming naively that the most efficient technology will prevail. However, even as we use our QWERTY keyboards, we realize that inefficient technologies persist despite our knowledge of more efficient alternatives (David, 1985).

The keyboard illustration underscores the taken-for-granted nature of technologies where institutionalized rules, social customs, and powerful elites shape their evolution (Powell, 1991). A similar realization of the taken-for-granted facets of organizational life led to the enunciation of institutional theories in organization studies (DiMaggio & Powell, 1983; Meyer & Rowan, 1977; Scott & Meyer, 1983; Zucker, 1977). These theories noted that organizations confront either technical or institutional environments, each exerting different types of pressures.

Upon reflection, institutional theorists realized that viewing organizations as confronting either a technical or an institutional environment at any given time creates a false dichotomy. Instead, by cross-classifying these two dimensions, they offered a typology in which certain organizations may be subject simultaneously to strong technical and institutional pressures. In particular, Scott and Meyer (1983) proposed that contradictory demands placed by these environments on organizations would lead to higher levels of internal conflict. They also proposed that organizations dealt with this dialectic tension by adopting more complex administrative systems, such as the matrix organizational form.

This dialectic view, however, falls short of an appreciation of the dualistic relationship between the technical and institutional environments confronted by organizations. For instance, Orru, Biggart, and Hamilton (1991) challenge Scott and Meyer’s (1983) argument that simultaneous technical and institutional demands promote internal conflict. Instead, they suggest that institutional pressures themselves are essential for the emergence of market order. Indeed, Powell (1991) and Dobbins (this volume) argue that institutional environments set the very criteria against which technical efficiency is judged (also see Barley, 1986, and Orlikowski, 1992, for related discussions).

It is this dualistic relationship between the technical and institutional environments that we explore in this chapter. Specifically, we focus on the evolution of unbounded technological systems—multicomponent systems that can be linked together in network industries (Garud & Kumaraswamy, 1993a). Our core proposition is that we can understand and better shape the evolution of technological systems by coupling their institutional and technical environments (which we liken to the two faces of Janus). This dualistic relationship between the institutional and technical environments creates a dynamic setting in which it is difficult for a dominant design to emerge. Therefore, we offer the notion of transient designs—designs that serve as stepping-stones for future product offerings.

Our chapter is organized as follows: In the next section, we provide an introduction to the technical and institutional environments of technological systems. Scott and Meyer (1983) define technical environments as those in which a product or service is produced and exchanged in a market such that organizations are rewarded for effective and efficient control of their production systems. Institutional environments are those character-
ized by the elaboration of rules and requirements to which individual organizations must conform if they are to receive support and legitimacy. In a similar manner, we suggest that the technical environment of a technological system consists of innovations and performance enhancements at the product level. The institutional environment of a technological system consists of a mosaic of interface and performance standards that together constitute the architecture of the technological system. These architectural standards prescribe the rules of engagement between system components and the criteria by which functional efficiency of the technological system is interpreted (Powell, 1991, p. 186). In this way, architectural standards are to technological systems what institutional environments are to organizations. We argue that there is a dualistic interaction between activities in these technical and institutional environments in several network industries. To illustrate this interaction, we explore dynamics associated with the evolution of Reduced Instruction Set Computing (RISC) systems in the workstation market of the computer industry. In the discussion and conclusion section, we explore the implications of these dynamics for theory and practice.

**Conceptual Framework**

Technological systems consist of a set of components that together provide utility to customers. System performance is dependent not only upon the performance of constituent components but also upon the extent to which they are compatible with one another (Gabel, 1987, p. 93; Henderson & Clark, 1990; Tushman & Rosenkopf, 1992). Compatibility between system components may be achieved by designing to a common set of standards. Standards are codified specifications that prescribe rules of engagement among components. Together, specifications about the form and function of components and the rules determining interaction among them define a system’s architecture.

**TECHNOLOGICAL DICHOTOMY AND DOMINANT DESIGNS**

The evolution of a system’s components and its overall architecture is best captured by the notion of technological trajectories (Dosi, 1982). Technological trajectories represent progression paths of technological systems in directions determined by previous choices (Powell, 1991). Trajectories also are determined by current perspectives of what is possible and feasible with a particular approach (Nelson & Winter, 1982).

During early stages of a system’s evolution, several technological trajectories might exist, each with its own system architecture. Components that perform well together as a system under one architecture might be incompatible with those that perform well under another architecture. To this extent, each architecture represents a unique configuration of components. As each architecture’s overall performance is determined cumulatively by the dimensions of merit of individual components comprising it (Tushman & Rosenkopf, 1992), different architectures will have different dimensions of merit.

During this “era of ferment” (Tushman & Anderson, 1986), there is little agreement on the rules of engagement and criteria on which performance should be measured. Therefore, each technological trajectory requires the creation of a distinct institutional environment that includes both the rules of engagement and the measurement criteria (Constant, 1980). Once created, each institutional environment forms the basis for stable expectations among mutually interdependent firms, thereby fostering complementary innovations and product refinement.

However, these very institutional environments constrain the development of each trajectory. Garud and Rappa (1994) illustrate how these institutional environments prescribe boundaries for future exploration, rendering researchers blind to the virtues of alternative trajectories as they ignore, deny, or distort performance reports that are not consistent with their own testing routines. Moreover, as these trajectories progress over time, researchers and firms develop idiosyncratic competencies that lock them into particular trajectories (Arthur, 1988; Cohen & Levinthal, 1990; David, 1985). Unable to redirect efforts toward pursuing alternate trajectories, rivals compete to shape emerging architectural standards that eventually congeal into a dominant design (Anderson & Tushman, 1990; Utterback & Abernathy, 1975).

Over time, architectural standards defining the dominant design become taken for granted as we use them routinely to prescribe which components (and consequently which firms) can interact, and which criteria should be used to evaluate system performance. Representing very high degrees of social order and cognitive construction (Jepperson, 1991) and prescribing the very basis for technological reality, this dominant architecture is to technological systems what institutions are to organizations. Gradually, an “inversion” occurs (Latour & Woolgar, 1979, p. 240), wherein rules of engagement and evaluation routines themselves become the basis for technological reality and competition during the era of incremental change. In this manner, the locus of interfirm competition shifts from the institutional to the technical environment after the emergence of the dominant design.
TECHNOLOGICAL DUALITY AND TRANSIENT DESIGNS

Though this two-stage model has served us well, its applicability in contemporary markets characterized by continual technological change must be reconsidered. In these continually changing markets, it is not possible to wait for a dominant design to emerge before proceeding to compete on technical merits. Nor is there any cessation of changes in the institutional environment of architectural standards. Instead of evolving through two distinct stages—one characterized by competition to shape institutional environment and the other by competition in the technical environment—these markets are characterized by simultaneous competition in both environments, one shaping the other in a reciprocal manner (see Van de Ven & Garud, 1994). In other words, these markets are characterized by dualism instead of a dichotomy.

We explore this dualistic process of technology evolution in network industries. Network industries consist of interrelated markets, each producing components of a larger technological system (Garud & Kumaraswamy, 1993a; Langlois & Robertson, 1992). These industries are also characterized by network externalities—situations in which the benefits a user derives from a product increase as others also use compatible products (Farrell & Saloner, 1986; Katz & Shapiro, 1985). For instance, the benefits derived by an individual subscribing to a data-sharing network increase with the number of other individuals subscribing to that network. Similarly, a complementary product, such as computer software, becomes cheaper (or more readily available) as the size of the network increases. Besides these direct benefits, indirect benefits of belonging to a large network include improved quality and lower price of after-sales service.

Several forces have led to a confluence of the institutional and technical environments in such industries. For instance, in the workstation market of the computer industry, advances in microprocessor technology and digital switching have led to connectivity between systems manufactured by different firms, thereby creating unbounded systems. Benefiting from the widening of their network boundaries, customers are demanding that firms offer systems that conform to industry-wide standards. In response to customer demands, manufacturers have begun offering systems that conform to industry-wide standards that no one firm controls. In such a setting, firms compete to shape the emerging institutional environment of architectural standards. One approach is to sponsor their technologies by licensing them freely to others, thereby creating an “open” architecture. Customers are attracted to networks built around these open architectures because open architectures promote compatibility among systems manufactured by different firms. As the size of the open network increases, so do benefits to customers and the viability of systems belonging to that network.

Open architectures that emerge through the sponsorship of standards foster competition in the technical environment as rivals offer alternative systems employing the same architectural standards. Therefore, even as firms shape architectural environments by sponsoring their technologies, they have to compete in their technical environments by innovating continually (see also Meyer & Rowan, 1977). Innovation in the product market, however, has to occur within the architectural umbrella that has been established so far. Otherwise, new product offerings will destroy earlier designs—a cannibalization process that is difficult to sustain indefinitely for both customers and manufacturers. At the same time, these product innovations extend and modify the very institutional environment that constrained and shaped them.

This interaction between the technical and the institutional environments represents a process of dualistic change wherein firms’ activities can be likened to rebuilding a ship (the architecture) plank by plank (through product innovations and enhancements) even as it sails. In such a setting, we suggest that products must be designed for transience, wherein any product is but a stepping-stone for future products. Such transient designs (Garud & Kumaraswamy, 1993b) lie at the nexus of the interactions between the technical and institutional environments. Because they conform to architectural standards, transient designs can be upgraded easily to offer new performance dimensions by rendering some facets obsolete, retaining some, and extending yet others. At the same time, innovations at the product level extend and reshape architectural standards.

In the next section, we illustrate this dualistic process of technology evolution and change in the workstation market of the computer industry. The computer industry was not always characterized by dualistic change. In the 1950s, during early stages of its evolution, the computer industry was characterized by product competition with an absence of architectural competition (Quadrant 1 of Figure 10.1). Several rival firms such as IBM, Control Data, Sperry, and Philco were active in the mainframe market, offering stand-alone systems that were comparable in functionality (Flamm, 1988, pp. 102-105). Firms could not shape industry-wide standards because networking capabilities were neither well developed nor considered important.

In the 1960s, when IBM’s System 360 architecture became dominant, the industry evolved to Quadrant 2, with an absence of both product and architectural competition. Product competition was weak because IBM’s large installed base created “transient incompatibility” costs (besides switching costs) for customers if they migrated to a system with a different
By the early 1980s advances in microprocessor technology and digital switching facilitated networking among computer systems, thereby paving the way for the introduction of workstations and the realization of Quadrant 4 dynamics (Figure 10.1). We explore these dynamics by examining events in the workstation market. To simplify our illustration, we focus on the effect of technological changes in one of the many components that constitute the workstation's overall architecture: its microprocessor.

### Evolution of RISC Microprocessor Systems in the Workstation Market

At the broadest level, the functionality of a microprocessor can be improved by increasing its speed or modifying its architecture. Since 1978 computer engineers have increased microprocessor speeds by a factor of eight, these advances coming from improvements in semiconductor materials and IC fabrication technologies (Tourma, 1993). However, there are limits to increasing speed by these methods. Aware of these limits, engineers have attempted to improve functionality by modifying microprocessor architecture.

A microprocessor's architecture depends on the instruction set that it uses. Traditionally, microprocessors have been designed using Complex Instruction Set Computing (CISC). CISC uses a comprehensive and lengthy set of instructions that requires greater microprocessor execution time. CISC chips have to be designed with duplicate circuitry to compensate for slower execution times, thereby increasing their size and complexity. An alternative to CISC is to employ a smaller instruction set that contains only those instructions that are used frequently. This approach is called Reduced Instruction Set Computing (RISC). RISC chips are faster because they have to execute a smaller instruction set ("A Risky New Architecture," 1985).

Despite its apparent superiority, RISC encountered resistance from proponents of CISC. Citing trade-offs between the two architectures, proponents of CISC questioned the very basis for RISC's superiority. They pointed out that performance measures employed to promote RISC, such as millions of instructions executed per second (MIPS) and millions of floating point operations completed per second (MFLOPS), were misleading. Specifically, CISC proponents argued that it was meaningless to compare the MIPS rating of systems with different instruction sets because the metric itself depended on the instruction set used (Hennessy & Patterson, 1989).
Of greater importance than the number of instructions processed per second, according to proponents of CISC, was the time taken by the system to execute typical programs. By this measure, CISC could perform as well as RISC. Because the CISC instruction set is elaborate, a CISC instruction can perform the task of several RISC instructions. Therefore, a typical software program written for CISC has fewer instructions to execute than one written for RISC and can run as rapidly (Touma, 1993). Thus, as with any new technology, there was an extended debate over the relative merits and performance measurement criteria of RISC and CISC. These debates led industry analysts to conclude that RISC would not become a commercial threat to CISC in the foreseeable future (“A Simpler Path,” 1986; “Gambling on RISC,” 1986; “High RISC Factors,” 1986).

Established computer manufacturers were reluctant to use RISC for fear of cannibalizing their traditional markets built around CISC systems. For instance, despite pioneering research on RISC in the mid-1970s (“Towards Faster,” 1985), IBM was reluctant to use RISC architecture in its computers for fear of cannibalizing its CISC-based mainframe products (Ferguson & Morris, 1993; “Gerstner’s New Vision,” 1993). Intel and Motorola, both dominant CISC microprocessor manufacturers, refused to initiate research on RISC, pointing to the adequacy of CISC architecture for commercial applications (“How Intel,” 1987). Of the established computer manufacturers, only Hewlett Packard (HP) pursued the RISC architecture and, of the established microprocessor manufacturers, only National Semiconductor began experimenting with RISC (“Gambling on RISC,” 1986). As is the case with the introduction of many new technologies, it was left to peripheral actors within the computer industry to commercialize the new RISC architecture.

RISC IN THE WORKSTATION MARKET

As in the wider computer industry, CISC microprocessor architecture was well established in the workstation market. Apollo pioneered the workstation market in 1981, with its proprietary CISC-based systems. Sun Microsystems also entered this market, in 1982, with workstations based on CISC microprocessors, but followed an unconventional open systems philosophy. In essence, this philosophy was to develop systems using standard, off-the-shelf components, and to license proprietary innovations to all firms without restriction (see Garud & Kumaraswamy, 1993a).

In 1985 William Joy of Sun Microsystems concluded that only RISC-based systems could meet customers’ demand for more power and functionality. Intel and Motorola, the dominant microprocessor manufacturers, spurned his suggestion that they develop RISC microprocessors. Subsequently, Sun Microsystems began efforts to develop its own RISC microprocessor in collaboration with Fujitsu. By 1987 Sun completed development of Scalable Processor ARCHitecture (SPARC), its own RISC architecture. Sun also announced the Sun-4 line of workstations based on its powerful SPARC architecture.

Consistent with its open systems philosophy, Sun announced its intentions to license SPARC to microprocessor manufacturers and rival computer manufacturers without restriction. In response to concerns about its continued intention to maintain SPARC as an open architecture, Sun, in 1989, created SPARC International, a trade group to promote SPARC. The mandate for SPARC International was to both ensure unrestricted licensing of SPARC technology and develop standards to ensure compatibility among different SPARC implementations.

Sun’s actions can be understood only in the context of important changes occurring in the workstation market and the computer industry at large. Advances in networking technologies allowed computer systems to be linked, thereby paving the way for distributed computation. Clearly this required that computer systems be built keeping multivendor compatibility issues in mind. Moreover, with the introduction of the IBM-PC in 1981, customers experienced the benefits of open systems built with standard, off-the-shelf components. Over time, they began demanding greater compatibility among systems offered by computer manufacturers throughout the computer industry—especially in the workstation market, where distributed computing and networking were important.

Opening up an architecture encourages decentralized innovation as licensee-firms innovate to a common architectural standard. Also, conformity to an open architecture increases compatibility among systems and components offered by licensee-firms. Eventually, an open architecture leads to unbundling of system components and the creation of a competitive market in complementary components. Instead of being locked into bounded networks of firms that restrict access to their proprietary technologies, customers enjoy the flexibility of mixing and matching compatible components to suit their specific needs. In addition, users find it easier to link disparate systems together in networks and enjoy the benefits of belonging to a growing, unbounded network. Thus, opening up an architecture makes it easier to mobilize widespread support among customers, potential rivals, and manufacturers of complementary components.

Though an open systems approach helps a firm mobilize support for its architecture, it also exposes the firm to competition in both the institutional and technical environments. In the institutional environment, the firm has to compete with rivals to establish its own system architecture as the architectural standard. At the same time, it has to compete in the technical environment with licensee-firms that offer compatible systems and components conforming to its architecture. Events subsequent to
Sun’s introduction of SPARC architecture illustrate dynamics that result from simultaneous competition in institutional and technical environments.

ARCHITECTURAL COMPETITION AMONG RISC-BASED WORKSTATIONS

Sun’s strategy of licensing SPARC without restriction had the intended consequence of mobilizing support for SPARC architecture, and for RISC-based systems in general. Initially, other workstation manufacturers responded to Sun’s SPARC systems by offering more powerful CISC-based workstations. As acceptance of SPARC architecture grew, rival workstation manufacturers became concerned that Sun would transform itself “into a shark trying to force its whole architecture down our throats” (MIPS’s John Mashey, quoted in “The Revolutionary,” 1988, p. 1).

Rival workstation manufacturers introduced their own RISC-based systems to prevent Sun’s SPARC from becoming the standard architecture for RISC-based systems. HP introduced workstations based on its PA-RISC microprocessor architecture, and IBM introduced workstations based on its own POWER RISC architecture. Some firms that had waited too long to make the transition from CISC to RISC introduced workstations built around RISC microprocessors supplied by third-party vendors. For instance, Digital Equipment Corporation (DEC) introduced workstations based on MIPS’s R2000 RISC architecture.

Dominant microprocessor manufacturers (Intel and Motorola) perceived these events as direct threats to their markets and initiated development of their own RISC architectures. Motorola introduced the 88000 RISC microprocessor. Intel, in a halfhearted attempt to enter the RISC microprocessor market, hurriedly repositioned 80386, a RISC coprocessor design, as a full-fledged microprocessor (“The Microprocessor,” 1989). Meanwhile, MIPS Computer Systems, one of RISC’s pioneers, promoted its own Rx000 RISC architecture by forming alliances with workstation and component manufacturers like Silicon Graphics, DEC, LSI Logic, and Siemens.

Appreciating the significant benefits of open architectures, these firms also initiated efforts to open their microprocessor architectures. Motorola established 88 Open Consortium Limited, a consortium to establish 88000 standards (“RISC Chip,” 1988). DEC developed its 64-bit Alpha RISC microprocessor and offered to license it to all manufacturers without restriction (“Digital Unveils,” 1992). Silicon Graphics, a niche player, acquired MIPS Computer Systems and undertook to keep MIPS’s RISC architecture open (“SGI Vows,” 1992). HP made a limited attempt to open up its PA-RISC architecture by forming Precision RISC Organization (PRO), a tightly knit nine-member association. However, HP decided not to license its RISC architecture broadly to prevent excessive competition from compatible systems (“HP Cathars,” 1992).

PRODUCT COMPETITION WITHIN SPARC ARCHITECTURE

Every step to promote an architecture as the industry standard by making it open means a step toward more intense competition in product markets. In the market for SPARC-compatible workstations, Solbourne Computer introduced its Series4 line of SPARC-compatible workstations in 1988. Immediately thereafter, Solbourne announced the release of its next generation of workstations—the Series5—based on Cypress’s speedier SPARC microprocessor (“Series 5 Caches,” 1989). Elsewhere, Sony and several firms from the Far East entered the workstation market with SPARC-compatible systems (“Taiwan’s PC Makers,” 1991). In 1991 alone, more than six different manufacturers introduced SPARC-compatible systems (“Six SPARC-Based,” 1991). These SPARC-compatible systems typically offered better performance at prices comparable to (or lower than) Sun’s own systems (“A Raft of Sun Clones,” 1991). For instance, Solbourne offered multiprocessing capabilities in its systems—features not offered by Sun—and cut prices by as much as 50% to make these the lowest priced workstations in the market (“Solbourne Targets,” 1991).

As Ferguson and Morris point out (1993, p. 143), opening up a static architecture is “merely giving the business away to clones.” Firms that open up their architectures can survive only by innovating continually. Indeed, Sun stayed one step ahead of rivals such as Solbourne only by “being nimble than its competitors in bringing out an endless succession of software and hardware products.” Sun introduced new families of SPARC systems with improved performance-to-price capabilities—SPARCstation1, SPARCstation2, and the easily upgradable SPARCstation 10. Further, Sun initiated development efforts with Texas Instruments (TI) and Fujitsu to create future-generation SPARC chips. Sun also introduced several software products that improved compatibility of its SPARC systems with rival workstation manufacturers and made it easier for users to network disparate systems.

Competition among SPARC-compatible systems demonstrates the dualistic interaction between the technical and institutional environments. In the case of an open architecture like SPARC, standardized specifications are transparent to all licensee-firms, so licensees can innovate within the umbrella of the open architecture. As licensee-firms innovate, they have to integrate these innovations into the architecture to maintain compatibility. Over time, the architecture itself changes as these innovations become assimilated.
Institutional Effects on Industries

Beyond the Race

Sun and its licensees firms created the SPARC International consortium, keeping these very dynamics in mind. As Sun and SPARC licensees introduced innovations, the original SPARC architecture, SPARC International developed standards to ensure compatibility among different implementations of SPARC. Slowly, the original SPARC architecture extended to SPARC International, and enhanced it further as they attempted to build systems with greater functionality and enter new markets ("Developers’ Open Race", 1992).

DUALISTIC CHANGE AND TRANSIENT MARKETS

The above descriptions illustrate how dynamics in the workstation market, where innovations in architecture and technology create change in both the institutional and technical environments, make it difficult for a dominant design to emerge. When designs all evolve constantly, firms have to work with transient designs that are not available in a variety of substrate materials. This means that the SPARC architecture can be enhanced as new materials occur (Giddens, 1992). In such an environment, it is difficult to predict the nature of the markets that will emerge. The SPARC architecture well illustrates the notion of a constant flow of new products. According to analysts, SPARC is the only technological architecture to develop a complete processor family that is compatible with existing hardware and software. SPARC has also been a major influence on other computer processors, and is expected to become a standard for high-performance computers and workstations. The SPARC architecture is also expected to support new markets, including supercomputers and personal computers. There is an additional layer of complexity that makes it difficult for a dominant design to emerge. As different markets and their corresponding architectures converge, "technology "transmission" occurs (Davidov & Malone, 1992). The resulting competition between RISC and CISC systems is expected to drive innovation and competition, leading to the development of new markets for both technologies. Meanwhile, Microsoft's success in integrating Windows NT and other multitasking capabilities into the SPARC architecture, made it a viable alternative to UNIX. Realizing the threat that Microsoft's action posed to their competitive positions, several workstation manufacturers, including Sun and HP,
These innovations occur within the confines of existing architectural standards that enable networking and compatibility. Working within architectural standards, rather than attempting to redefine them, allows firms to innovate and compete without having to develop new interfaces or protocols. This approach reduces the risk of investing in technologies that may not gain widespread adoption.

In these environments, innovation is not just about creating new products or services, but also about finding new ways to use existing technologies. For example, firms may develop new algorithms that improve the performance of existing computer architectures. These improvements can be used by other firms that are building on the same technology, thereby spreading the benefits throughout the industry.

This approach also helps firms to manage the complexity of their operations. By working within established standards, firms can allocate resources more effectively and reduce the risk of compatibility issues. This allows them to focus on other aspects of their business, such as customer service and product development, without having to worry about the technical details of their infrastructure.

In summary, the use of existing architectural standards is a key factor in the success of innovation in the computer industry. By leveraging existing technologies and building on them, firms can create new value and compete more effectively. This approach is essential for the continued growth and development of the industry.
Notes

1. Tushman and Rosenkopf (1992) label these as "open systems," a term that we reserve to denote systems that are built using standard, off-the-shelf components and, therefore, have open architectures.

2. In the case of computers, standards would have to specify "how programs and commands will work and data will move around the system—the communication protocols that hardware components must follow, the rules for exchanging data between application software packages and the operating system, the allowable font descriptions that can be communicated to a printer, and so forth" (Ferguson & Morris, 1993, p. 120).

3. Other complementary approaches to shape architectures include (a) co-optation of institutional bodies (Hirsch, 1975), (b) lobbying to shape emerging regulation (Leone, 1986), and (c) the propagation of a standard through committees (Farrell & Sabel, 1988).

4. The computer industry traces its origins to military applications, with the government as its main sponsor. By the late 1940s (the point of entry for us in this chapter), computer applications had diffused into the commercial sector.

5. It is important to distinguish between our classification and the one offered by Scott (1992, p. 133). Scott demarcated four different contexts based on the strength (or weakness) of the technical and institutional environments that organizations confront. In this chapter, we are operating in a context where both technical and institutional environments are strong. Given the presence of strong technical and institutional environments, our focus is on whether there is competition within each of these two environments.

6. Other core components that determine a workstation's architecture include its operating system, memory, systems bus, and networking hardware.

7. After RISC became a commercially viable alternative to CISC, several benchmark measures came into vogue to make valid performance comparisons between CISC-based and RISC-based systems. Over time, SPBC (Systems Performance Evaluation Cooperative) gained wide acceptance as the leading benchmark to measure system performance (Weicker, 1990).

8. By the end of 1990, Sun's SPARC architecture commanded a 35% share of the workstation market, with MIPS (24%) and Motorola (12%) being the other major players ("RISCy Business," 1990).

9. In 1992 Sun's SPARC systems accounted for 38% of the workstation market, with HP (17%), DEC (12%), and IBM (7%) being the other major players. Still, no single architecture was dominant.

10. This observation is well illustrated by IBM's experience in the PC market, where IBM's inability to innovate continually and keep up with clone makers cost it market dominance.

11. A group of IBM-compatible PC manufacturers led by Compaq—the ACE Consortium—endorsed a partial shift from Intel's 80x86 CISC microprocessors to powerful RISC microprocessors designed by MIPS Computer. Intel quickened its product development efforts to create CISC microprocessors like 80486 and Pentium that were comparable to RISC's functionality. Citing its product development efforts, Intel persuaded Compaq to stay with its CISC microprocessors, and, to an extent, neutralized the threat of large-scale defection of PC manufacturers from its CISC microprocessors to RISC ("Compaq Emerges," 1992). Apple and IBM, two firms that had suffered loss in market position in the PC market, decided to collaborate with Motorola to develop a new RISC microprocessor called PowerPC. They also announced their intention to make the PowerPC microprocessor an alternative open standard to Intel's CISC microprocessors ("Apple/IBM Birthday," 1992).

Institutional Interpretations and Explanations of Differences in American and Danish Approaches to Innovation

PETER KARNOE

This chapter argues that the sharp distinction between institutional and technical environments embedded in the earlier versions of organizational institutionalism has detracted attention from the fact that even the social organization of technical practices is institutionally shaped. Technical practices are exposed to strong legitimate, coercive, and normative pressures, with social rules and norms strongly guiding acceptable and appropriate ways of organizing economic activities. In this sense, technical practices may be seen as institutionally constructed social practices.

I support this argument with empirical research on the different innovative practices associated with the communities of Danish and American entrepreneurs that created modern wind power as an energy technology. This study covers the emergence and early consolidation phase of modern wind turbine technology from 1974 to 1990. Actors in both countries were supported by government market subsidies to stimulate the installation of wind power and by a test and research center designed to support industrial technological development. I will attempt to demonstrate how the different

Author's Note: The initial research for this chapter was made while the author was a visiting scholar at the Department of Economics, Stanford University. This stay was supported by the Danish Social Science Research Council and made possible by Professor Nathan Rosenberg. The author has in different phases benefited from comments by and encouragements from Frank Dobbin, Raghu Garud, Dennis Gioia, James H ö pner, Kristian Kreiner, Peer H. Kristensen, James March, Nathan Rosenberg, Dick Scott, and Ed Steinmetz.
The Institutional Construction Of Organizations
International and Longitudinal Studies

W. Richard Scott Søren Christensen
Editors

1995

SAGE Publications
International Educational and Professional Publisher
Thousand Oaks · London · New Delhi