European Management Journal (2009) **27**, 281–292

journal homepage: www.elsevier.com/locate/emj

Modularity vs programmability in design of international products: Beyond the standardization—adaptation tradeoff?

Mark Lehrer *, Michael Behnam

Associate Professor of Management, Sawyer Business School, Suffolk University, 8 Ashburton Place, Boston, MA 02108-2770, United States

KEYWORDS
Design innovation;
Modularity;
Programmability;
European market diversity;
ERP software;
Multinational corporations;
SAP;
Standardization—adaptation

Summary This paper argues that *design innovation* constitutes an important means for multinational corporations (MNCs) to manage the standardization—adaptation dilemma in product development for international markets. Through design innovation MNCs can accommodate the heterogeneity of different national markets by building functional versatility into products. Two design principles are covered. First, the familiar design principle of modularity is reviewed along with its bearing on the standardization—adaptation dilemma. Second, the less familiar design principle of programmability is introduced. A historical case study on ERP (enterprise resource planning) software is presented, showing how one firm (SAP AG) used design innovation to serve internationally heterogeneous markets. The study also illustrates the interplay of these two principles; in essence, modularity helps reduce some of the complexity costs that arise from increasing levels of programmability in product design.

Introduction

A central preoccupation of research on multinational corporations (MNCs) concerns the tension between the heterogeneous conditions of the MNC's markets on the demand side and the need for production efficiency on the supply side (Kotabe, 2003). Although producing and selling on a global scale can provide MNCs with supply-side economies, the diversity of national markets on the demand side — a consistent challenge for European firms — inherently interferes with production and marketing efficiency. Since the 1970s a major research paradigm has therefore centered on the standardization—adaptation tradeoff in product development for international markets (Rau and Preble, 1987; Samiee and Roth, 1992; Theodosiou and Leonidou, 2003). Notwithstanding proclamations that globalization would lead to worldwide standardization of products (Levitt, 1983), the more widely accepted view is that MNCs have to find the proper ''balance'' between standardization and adaptation in developing products for international markets (Subramanian and Hewett, 1993). In a related vein, Bartlett and

* Corresponding author. Tel.: +1 617 573 8000.
E-mail addresses: mlehrer@suffolk.edu (M. Lehrer), mbehnam@suffolk.edu (M. Behnam).

0263-2373/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.emj.2009.01.003

Ghoshal's (1989) work on transnational organizations examined the way in which MNCs adjust their strategy and organization to cope with the competing imperatives of global integration (standardization) and local responsiveness (adaptation).

To date the balance between standardization and adaptation in product development has been primarily conceptualized and measured as a tradeoff (Theodosiou and Leonidou, 2003). The present research therefore addresses methods by means of which MNCs can aim to *transcend* this tradeoff. One of those means — and the one explored in this research — concerns the use of innovative *design principles*. The broader research question addressed here can be formulated as: ''How can MNCs use design principles to reconcile the objectives of standardization and adaptation in products, i.e. to achieve high levels of *both* standardization *and* adaptation?''

In the framework developed here, design techniques for achieving a higher-order combination of standardization and adaptation can be subsumed under the two distinct categories of *modularity* and *programmability*. Modularity in design involves decomposing a product into separable components. Modularity can reduce the amount of work needed to tailor products to local markets because it permits the standardization of selected components and the local adaptation of others. A prominent example is the modular design of cars which allows both the sharing of components across different product lines and adaptation to specific requirements in different country markets. Programmability, in contrast, reconciles standardization and adaptation by incorporating into products the ability to adapt to a multiplicity of market settings. For example, multiband cellphones and all-region DVD players can switch between different technical standards so that they can be operated around the world.

Discussion proceeds as follows. We begin by reviewing the standardization-adaptation debate, highlighting the limited scope accorded to product design issues. The next section presents a framework of modularity and programmability as design principles against the background of prior scholarship. Although much has been written on modularity and although much of this work is clearly relevant to the standardization-adaptation challenge facing international firms, the connection has rarely been made explicitly, with the result that modular design principles have received curiously little attention in research on MNCs. The design principle of programmability, though likewise common in the industrial practice of MNCs, has generally been little explored. The methodology and findings sections illustrate both design principles with a historical case study of SAP AG, examining how the firm dealt with the standardization-adaptation dilemma in its product development process. The ensuing discussion section formalizes an interesting pattern that emerges from the case study, namely that modularity in essence palliates the side effects of programmability: modularity helps reduce some of the complexity costs that arise from increasing levels of programmability in product design. The article concludes with economic and technological considerations as to why business application software is not an entirely idiosyncratic type of product but instead shares certain characteristics with a growing number of technology-intensive goods.

International product development in MNCs: the standardization-adaptation debate

Traditionally MNCs are seen to confront the conflicting imperatives of global integration and local responsiveness (Prahalad and Doz, 1987), a tension conceived usually in organizational terms. For example, Bartlett and Ghoshal (1989) and Nohria and Ghoshal (1997) conceptualize the ''transnational'' MNC as a differentiated network in which dispersed, specialized and interdependent units rely on intensive intra-organizational knowledge flows in order to deal with the challenges of standardization and adaptation simultaneously. Product design, however, constitutes another fundamental means by which MNCs can reconcile the antithetical pulls of global integration and local responsiveness. The issue can be framed in terms of standardization (the design equivalent of global integration) and adaptation (the design equivalent of local responsiveness). Spurred on by Levitt's (1983) famous prediction of the advent of standardized global products, marketing scholars have studied the standardization-adaptation tradeoff in product development for international markets, with Rau and Preble (1987), Samiee and Roth (1992), and Theodosiou and Leonidou (2003) providing overviews of the standardization-adaptation debate.

Because most of the debate has been framed in terms of an either/or between the poles of standardization or adaptation, most contributors fall into three fairly predictable camps, namely globalization proponents, adaptation proponents, and contingency advocates (Theodosiou and Leonidou, 2003). Broadly speaking, basic findings about the net performance implications of standardization and adaptation are inconclusive (Rau and Preble, 1987; Zou and Cavusgil, 2002). There is comparatively greater agreement about certain contingency relationships. Industrial products represent better candidates for standardization than consumer products (Jain, 1989); within the consumer product segment, durable goods permit more standardization than non-durables (Baalbaki and Malhorta, 1993). Rapid changes in technology similarly favor standardization (Samiee and Roth, 1992).

In fact, product *design* issues per se figure only weakly in the standardization-adaptation discussion. Given the marketing orientation that has dominated this discussion, product attributes constitute only one aspect of the marketing mix along with promotion, pricing, and distribution decisions (Baalbaki and Malhorta, 1993). Since this research stream is concerned chiefly with capturing the multivariate nature of the standardization-adaptation tradeoff, innovations that seek to transcend the tradeoff in specific areas such as product design are only sporadically discussed.

Framework: two design principles contrasted

The subsequent framework and analysis is built upon the following basic assumptions. First, we postulate that MNCs can employ a wide range of innovative design techniques to transcend the standardization-adaptation tradeoff but that these various methods can be condensed down to a relatively small number of fundamental underlying *design principles*, used either separately or in combination. Second, we

assume that these design principles are not mutually exclusive and that any number of them may be potentially embodied in a given product. Because a combination of these principles is possible, design principles like modularity and programmability can be considered ''dimensions'' of design. Though the following analysis examines only these two dimensions of design, this is not to deny the existence of other design principles and other dimensions of design.

Figure 1 represents the two dimensions of design innovation examined in this research. Along the vertical axis, various degrees of product *modularity* are depicted. At the bottom of the axis, a fully integrated product, i.e. without modular design, is represented in which the choice between standardization and adaptation is a pure tradeoff. Yet as one ascends the vertical axis, i.e. employing modular design, it becomes possible to improve the terms of the tradeoff. *Platform modularity*, for example, involves separating the overall product into a standardized core platform on the one hand and a set of variable components on the other, including components that can be produced by external parties. The term ''platform modularity'' is derived from the method of using product platforms (Lehnerd and Meyer, 1997; Meyer, 1998) for churning out a variety of products rapidly and at low cost by having them share common components and processes. Takeuchi and Porter (1986) refer in a similar vein to ''modified global products'' with a standardized core platform permitting peripheral modifications that vary from country to country. Manufacturers of aircraft, trucks, and automobiles use platform modularity as a matter of course. Swan, Kotabe, and Allred (2005) noted Honda's use of a standardized platform to produce a variety of automobiles at comparatively low cost to cope with different markets and their evolution over time.

These economic advantages continue to apply as one moves up the vertical axis from platform modularity to *component modularity*. In such products the standardized core disappears and gives way to a product architecture of interchangeable components, as in a stereo system (Langlois and Robertson, 1995) or in a Wintel personal computer (Chesbrough and Teece, 1996). Here it is not the core product that is standardized but the interfaces between components (Miozzo and Grimshaw, 2005) or, as Langlois (2003, p. 374) puts it, ''the rules of the game'': ''By taking standardization to a more abstract level, modularity reduces ... the amount of product standardization necessary to achieve high throughput'' (Langlois, 2003, p. 375). Put differently,

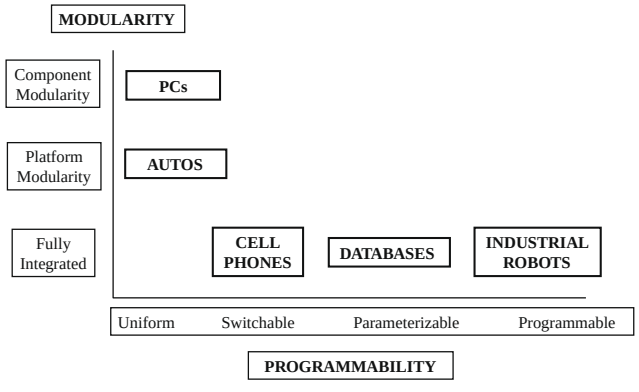

Figure 1 Design principles for global products.

modular design facilitates ''strategic flexibility'' (Sanchez, 1995) and ''economies of substitution'' (Garud and Kumaraswamy, 1995), allowing specific parts of the overall product to be upgraded or adapted to specific circumstances without having to redesign the product from scratch. Whereas platform modularity implies that a core firm dominates overall product development, a product architecture featuring component modularity need not necessarily be masterminded by a single firm; instead, production may be organized in a ''modular production network'' (Sturgeon, 2002). In this sense, product modularity can entail organizational modularity (Sanchez and Mahoney, 1996).

Modularity in design can help overcome the tradeoff between standardization and adaptation in both direct and indirect ways. The *direct* way is by reducing the cost of adapting a product to heterogeneous markets, as discussed above. Yet there is a second, *indirect* way that modularity can help overcome the tradeoff pertaining to the SAP case. Modularity reduces complexity for both producers and customers. As Baldwin and Clark (1997) explain, modular design involves the partitioning of production tasks into design visible rules and ''hidden information'' (i.e. the fact that producers and users need not worry about details of the whole product as long as they adhere to the design rules concerning each individual part), resulting in economies of information processing. Since a software product that is both standardized *and* adapted to multiple markets is apt to be highly complex — and in the case of SAP's enterprise resource planning (ERP) software is *extraordinarily* complex, further design improvements that lower the costs associated with complexity (''complexity costs'') can be critical to the product's diffusion.

To summarize the vertical axis, modularity is a well-established principle in manufacturing for reconciling scale economies with product variety (Hayes, Wheelwright, and Clark, 1988; Lehnerd and Meyer, 1997), yet remains a little researched topic concerning the specific problem of serving heterogeneous national markets. The same is undoubtedly true of programmability as well, the design principle depicted along the horizontal axis of Figure 1. Here the products concerned are chiefly those involving electronic circuitry. Programmability in design facilitates the inclusion of adjustable features in a product that allow it to adapt — manually or automatically — to local requirements and to function like a locally tailored product. A simple example of this is the 120/240 V AC adapter now featured in personal computers. AC adapters used to work on only a single voltage, thus requiring PC owners to possess multiple transformers to operate their PCs on different continents. However, design improvements now allow a single AC adapter to work universally. The design feature of programmability thus engenders a kind of product standardization that is not just a compromise among variants, but a built-in capacity of the product to adapt to different environments. Unlike modularity, programmability does not bifurcate the product into standardized and locally tailored components; instead, the product contains adjustable features allowing it to accommodate multiple environments.

Along the horizontal axis of programmability in Figure 1, AC adapters and cellphones occupy a relatively low level of programmability which we denote as ''switchable'': the product adapts to a new local setting either by itself or upon

the flick of a manual switch. As one progresses along the axis to higher levels of programmability, the product's capacity to adapt to more complex changes in its environment increases, but so too does the need for user intervention to manipulate the product's settings. Borrowed from the vocabulary of software programmers, the term ''parameterizable'' derives from the process of setting up software programs (''parameterization'') for use by specific users. Parameterization essentially involves answering a number of ordered questions about how the product will be used in its specific environment. For example, Windows NT is ''parameterizable,'' necessitating answers to a set of queries during set-up; at the same time, there is usually no need to engage in the actual software coding that a truly ''programmable'' product would entail (right-most side of the programmability axis).

A fully ''programmable'' product, such as a mainframe computer or industrial robot, may be completely standardized in terms of what comes in the box and yet still require programming by the user for it to become operational in a specific context. Programmability is the characteristic that allows such a product to be at once standardized (one size fits all) and adaptable to local conditions. However, the fact that it requires special programming by the user means that its versatility comes at the expense of greater complexity and implementation costs.

By way of summary, the effort to reconcile standardization and adaptation is akin to the goal of ''robustness'' in international product design (Swan et al., 2005). As defined originally by Rothwell and Gardiner (1984), a robust design encompasses both the product characteristics required in current markets as well as those that will be needed as market requirements evolve in the future with a view to optimizing development and production costs over time. In essence, ''robustness'' applies to the diachronic problem of designing products for temporally different environments, whereas the design of international products by MNCs involves a more synchronic version of the same problem applied to spatially distinct markets.

Methodology and data

Methodological approach

Empirical investigation took the form of an exploratory case study. Among the many uses of such a study (Yin, 1993), one involves the investigation of causal mechanisms that remain hidden because they apply to outliers and exceptions in a population rather than to the majority of cases. The standardized (i.e. non-customized) segment of the software industry represents a fertile ground for such a study. The reason is that it came to be dominated almost completely by US producers (Table 1). Among European software producers, SAP became the one and only leviathan in standardized software and has remained unique in this respect.[1]

[1] According to the Truffle 100 Survey, SAP ranked first in 2006 among European software companies in sales with 9.4 billion Euros in revenue, followed in second place by Sage and Dassault Systems with only 1.4 and 1.2 billion Euros in revenue, respectively.

While numerous factors play in favor of US software developers, one of the most important is considered to be the size of the domestic market in combination with the vast economies of scale inherent in software production (Mowery, 1996; Campbell-Kelly, 2003). Hence, one of the objectives guiding this research concerned the identification of contextual conditions and firm actions that could permit a software firm from a smaller country (i.e. Germany) to prevail in certain segments. In other words, the focus on ERP software and SAP AG was conducted against the background of contrasting software segments dominated by firms of US origin.

Consistent with this research background, the following sections report on two basic sets of findings: (1) the contextual, industry segment-specific conditions allowing a firm from a smaller (e.g. European) country to produce a dominant firm in the ERP software segment; and (2) more importantly, some of the specific actions taken by this dominant firm (i.e. SAP) to exploit these conditions. Since many of these actions involved design decisions and the innovative use of various design principles, the account given below stylizes this latter set of findings in terms of a generalized theoretical framework involving design principles. This framework is not inductive in the sense of being derived exclusively from the study. Instead, the study pursues the goal of ''analytic generalization'' (Yin, 2003), a theory-building process that uses previously developed findings and theory as a background against which to assess the results of the case study (Tellis, 1997). As Yin (2003, p. 38) felicitously expresses it, the objective of analytical generalization is not to generalize the case study to other cases, but to ''generalize findings to 'theory,' analogous to the way a scientist generalizes from experimental results to theory.'' The present effort at analytic generalization synthesizes specific findings on design innovations in ERP software with broader theoretical considerations on the use of design principles in international product development.

Data sources and triangulation

The case study was based on two main data sources: (1) primary research on the German IT sector, and (2) published interviews with the founder-managers of SAP. Primary research consisted of 23 interviews conducted in the German IT sector (summarized in Table 2) as part of an ongoing research project on the German IT sector (Lehrer, 2000; Lehrer and Asakawa, 2002; Lehrer, 2005). The interviewees included directors of two Fraunhofer Institutes specializing in IT, managers of three German software companies (SAP, IBM Deutschland, Nemetschek GmbH), and participants at two German software industry events. Among published interviews with top manager-founders of SAP, one source of information is pre-eminent. This is a book-length dialogue about SAP's development with Hasso Plattner, co-founder, main software developer and second CEO of the firm (Plattner, 2000); the dialogue is moderated by a leading German IT expert, Professor August-Wilhelm Scheer, thus enabling unusually deep exploration of technical issues in the firm's development. Since Plattner was an original founder of the firm in 1972 and a top manager for over thirty years, this source of information is given special weight.

Table 1 World's largest software companies by sales, 2007.

Rank	Company	Sales	Nationality
1	Microsoft	57.9	USA
2	Oracle	21.0	USA
3	**SAP**	**16.1**	**Germany**
4	Computer associates	4.2	USA
5	Adobe	3.4	USA
6	Electronic arts	3.2	USA
7	Amdocs	2.9	USA
8	Intuit	2.8	USA
9	Activision	2.6	USA
10	Autodesk	2.2	USA

Source: Reuters sales are USD billions.

Table 2 Overview of primary research.

German IT organizations	Nature of organization	Persons interviewed
SAP	Leading ERP software firm	Six managers of SAP and SAP customers at SAP customer event
Fraunhofer Institute for Computer Graphics	Software R&D Institute (Darmstadt)	Director of institute
Fraunhofer Institute for Software and Systems Engineering	Software R&D Institute (Berlin)	Director of institute
IBM Deutschland Development Center	Major IBM Software Development Center (Böblingen)	Three top research directors
Nemetschek GmbH	Largest German Provider of Architectural CAD Software (Munich)	Founder & head of firm; Sales director
BVIT	Industry Association for Independent German Software Companies	Director; Data analyst
CeBIT Fair	Largest Annual German IT Fair	8 representatives at 3 IT consulting companies installing standardized software (mainly pre-scheduled interviews, mainly with executives)

Basic findings

Research disclosed that a European firm could dominate the development of ERP software because, for one thing, US software developers saw little market for such a product. In essence, ERP software is an integrated suite of function-based business applications (production, finance, HRM, etc.) supporting the user firm's daily operations. Whereas SAP had developed cross-functionally integrated software since the 1970s, most software developers in the US considered such a product impractical, requiring excessive planning and involving too much inflexibility whenever changes were needed in individual functional components. Most US buyers and sellers of business software preferred to deal in function-specific products (production software, finance software, HRM software, etc.), with each of these segments dominated by a different set of firms. Obviously this raises questions such as: Why were German corporate customers installing integrated business software while US firms purchased business software using a mix-and-match approach? What was unique about the German software market? Why did this uniqueness eventually prove to be an asset rather than a handicap for SAP's international expansion? The next section summarizes the answers to these questions.

In a nutshell, SAP took on a standardization-adaptation problem that US software firms considered impractical even to attempt to solve. SAP used the design principles of modularity and programmability to engineer a cross-functionally integrated software suite for installation by users in heterogeneous industries and, subsequently, in heterogeneous countries. Despite its adaptability to different environments, SAP's product was so standardized that it came in only two versions: R/2 (for mainframes) and R/3 (for client-server networks). By the 1990s, when ERP software (especially R/3) had proven to be a feasible and useful tool for global firms to integrate their worldwide operations, SAP possessed a sizeable first-mover advantage over later ERP followers like PeopleSoft and Oracle.

In ameliorating the terms of the standardization-adaptation tradeoff through innovative product design, SAP's primary design principle was programmability. At the same time, SAP incorporated modularity into its product design as well, but for a different purpose: to reduce complexity, in particular the complexity arising from increasing programmability. The basic story line encapsulated in the fol-

lowing narrative is that SAP's use of programmability to resolve the standardization-adaptation tradeoff came at the cost of increasing product complexity and hence difficulty of implementation by users; to mitigate implementation costs, SAP incorporated modular features of design into R/2 and R/3. This narrative is intended to document a novel pattern in the interplay of markets and technology (Ettlie and Subramanian, 2004) rather than to furnish a prescriptive formula for the design of international products; indeed, the use of programmability as a design principle to address the standardization-adaptation problem entailed new tradeoffs.

Findings: the interplay of modularity and programmability

Overview of findings on SAP's markets and design innovation

SAP's particular use of design principles was strongly influenced by the different national contexts in which SAP developed its software product. The specific nature of demand in its home country, Germany, was propitious for the initial standardization and programmability of software design. Exposure to diverse foreign markets led to increasing levels of programmability in later stages of SAP's product development process. Both R/2 in the 1980s and R/3 in the 1990s made use of component modularity to mitigate the difficulty and cost of implementation associated with a product of increasing levels of programmability and hence increasing complexity.

As depicted in Figure 2, SAP's software packages were both parameterizable and programmable. An indication of the complexity associated with ERP software is that R/3 allows the client firm to fill in approximately 25,000 different tables to permit customization of the software to a given customer's premises. This process is actually called ''parameterization.'' Both R/2 and R/3 are programmable products as well, accompanied by a special SAP-developed programming tool to generate supplementary code to perform operations not covered by the basic package. This important programming tool is ABAP, a so-called fourth-generation programming language invented by SAP. For example, ABAP/4 is used both by SAP to develop R/3 modules and by SAP customers to carry out supplementary coding. SAP uses the slogan that ''R/3 is developed with R/3.''

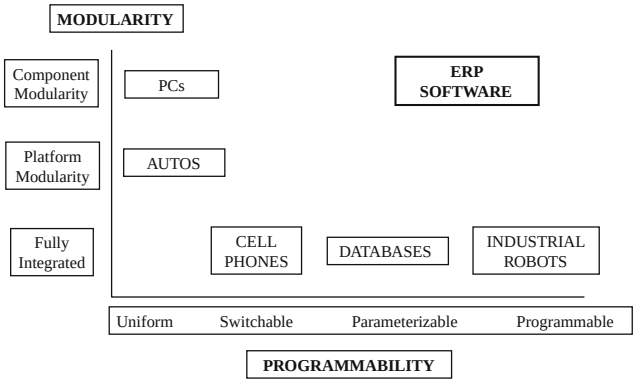

Figure 2 Combined design principles at SAP.

The following historic reconstruction organizes SAP's design innovations according to the key markets it served over time. Each major market provided a specific demand-side stimulus to innovation, obliging SAP to endow its software product with added design features. The basic findings concerning programmability and modularity are summarized in Table 3.

Stimulus to design innovation from key market #1: Germany

Founded in 1972, SAP specialized in business application software, i.e. software controlling the basic business functions of the company (accounting, finance, production, etc.). Business application software up to this time was typically not standardized but rather customized in accordance with the company's specifications. SAP began by writing customized software in this way, albeit with the strategy of recycling code as much as possible. By developing reusable code as work migrated from one customer to another SAP progressively enlarged the standardized core code with ever more adjustable settings and parameters over time; such programmability allowed the product to be installed by firms in almost any industry. A milestone was SAP's first entirely standardized package R/2 (introduced in 1981); this was later followed by its successor R/3 (1992).

R/2 diffused quickly throughout the German large-firm sector and by 1990 was installed by the majority of Germany's top 100 firms. For most of the 1970s and 1980s, SAP's ''integrated application software'' (rechristened as ''enterprise resource planning'' software in the early 1990s) was idiosyncratic and specific to the German market. Interviewees explained that cross-functionally integrated standardized software originally appeared feasible only in business environments with a high tolerance for detailed planning of business tasks and where business processes were relatively stable over time.

Germany offered such a business environment. Germany's idiosyncratic market for standardized business software was evidently driven by the high degree of business standardization in Germany. This point was made by several interviewees, but is also underlined in research on German business institutions. Beyond the famous system of DIN norms, industrial practices concerning training procedures, work processes and job classifications are extensively codified by powerful industry and labor organizations (Lane, 1989; Hall and Soskice, 2001). Cross-national comparisons disclose more work-process homogeneity among German firms than among comparable US or UK firms (Lane and Bachmann, 1996; Grimshaw and Miozzo, 2006). Even major IT decisions by firms in Germany involve greater interfirm standardization of processes, driven in part by substantial deliberation between management and works councils (Miozzo and Grimshaw, 2006).

While R/2 provided integration across firm functions to replace the characteristic patchwork of function-specific IT systems in client firms, it raised the level of difficulty, cost and risk involved in supplanting all of the client firm's older IT systems at once. R/2 palliated these problems

Table 3 Design principles in product development at SAP.

Key market, time frame	Extension of application range	Application of design principles
Germany, 1970s–1980s	Across different firms and industries	*Programmability* of software product to enable client customization of a basic standardized code; *Modularity* of functions to permit stepwise installation of the ERP suite or only of selected parts
Europe, 1980s	Across national currencies & tax laws	*Programmability*: incorporation of parameters to facilitate operation in multi-language, multi-currency, multi-tax code environments
Japan, 1980s	Across languages & symbols	*Programmability* in graphical representation: double-byte version of R/2 (and later R/3) to accommodate non-Western characters
USA, 1990s	Across differing firm sizes and hardware platforms	*Programmability* of software product to permit installation on multiple hardware platforms; *Modularity* to enable scalability and installation on client-server systems

through design modularity. R/2 was built upon a set of functional components called modules that can be installed one at a time. In terms of Figures 1 and 2, SAP employed ''component modularity'' in design. By permitting sequential rather than simultaneous installation of components, modularity allows stepwise substitution of legacy IT systems. A list of the core R/2 modules is included in Table 4.

All of this poses an obvious puzzle. Although Germany was an idiosyncratic market for cross-functionally integrated business software in the 1970s, it did not remain so. In the 1990s the concept of cross-functionally integrated business software caught on worldwide:

> Even in the 1990s the concept of integrated software was a new idea in the US. It was also new for the rest of the world, although it was already familiar in Germany in the 1970s. We had introduced the concept of ''integrated application software,'' and it was difficult to sell non-integrated application software in Germany, be it SAP software or software from rival firms. But America had separate software components with a batch interface (Plattner, 2000, pp. 38–39).

Two major interrelated reasons developed below explain why SAP's software evolved from a niche to a mainstream software product. The first was cumulative diffusion of the software among customers. As R/2 dominated the German market for business software on mainframes in the 1970s and 1980s, foreign firms began to take notice, especially multinational firms that were impressed with the R/2 installations in their German subsidiaries. The second reason was continuing design innovation that expanded the application range of the software. In particular, increasing versatility incorporated into SAP's software package allowed the product to function in an ever wider range of national environments. This versatility was achieved along the design dimension of programmability.

Table 4 List of R/2 modules.

RF	RA
Financial Accounting	Assets Accounting
RK	RK-P
Cost Accounting	Projects
RP	RM-INST
Human Resources	Plant Maintenance
RM-QSS	RM-MAT
Quality Assurance	Materials Management
RM-PPS	RV
Production Planning and Control	Sales and Distribution

Stimulus to design innovation from key market #2: Europe

Whereas the German home base specifically encouraged standardization and cross-functional integration, SAP's European base provided a stimulus to development of multi-country compatibility through inclusion of further settings and parameters (i.e. more programmability). In particular, the European base provided the impetus for the company to support multiple languages, tax laws, and currency in its software from comparatively early on, as co-founder Plattner emphasized:

> We supported multiple currencies. This is the great advantage for a European firm. And now [1999] we have another currency in parallel, the Euro. In America no-one understands that we have two currencies to cope with in each country. And we have multiple currencies in every country. This is a complexity that American software has trouble dealing with. We also have different tax laws in each country of the EU. We have authorities in Brussels, but there remain different tax laws and (national) variations, and the variations are much bigger than those in the US (Plattner, 2000, p. 49).

An explanatory hypothesis that crystallized in the course of interviews with German IT experts is that SAP's software design benefited from the contrasting environmental characteristics of its German and European environment. The German business context, through its environmental *uniformity* (i.e. standardized business practices), induced SAP to adopt a standardized software design that could perform business applications across a range of firms and industries (design principles: programmability and modularity). In con-

trast, the European business context, through its environmental *diversity*, impelled SAP to add parameters to its software products for adjusting to different business environments (design principle: programmability). In other words, SAP's product development leveraged the fact that it was simultaneously exposed to environments favoring a high level of standardization (Germany) and environments encouraging a high level of adaptation (Europe).

Stimulus to design innovation from key market #3: Japan

A significant design milestone resulted from the decision to support Japanese for R/2. This seemingly innocuous requirement in fact required SAP to develop a double-byte version of the entire software, the costliest engineering modification SAP had ever engaged in until then (late 1980s). The strategic consequence of this engineering change was that SAP's software could subsequently support other languages requiring double-byte characters, notably Chinese and Korean. In other words, adapting R/2 (and later R/3) for use in Japan did not entail just a translation of terms into Japanese, but rather an added level of programmability in graphical representation (double-byte character capacity), permitting the product to be implemented throughout Asia.

SAP's decision to support Japanese was prompted by wishes of its lead users, namely multinational firms. Although it required three years of programming effort, SAP took on the project because ''the German multinational chemical companies asked us to go to Japan, then DuPont asked us to go as well'' (Plattner, 2000, p. 48). The strategic significance of language flexibility extended well beyond just the need to serve Asian markets. It was the ability of the software to function in Asian *as well as* in Western countries that enabled SAP's software to be utilized as a platform for integrating the cross-border operations of multinational companies. SAP had not foreseen this use of its software and such a development was unforeseeable throughout the 1980s, as long as national subsidiaries of global firms continued to have their own separate mainframe systems with only weak cross-border IT interconnectivity. This changed in the 1990s with the advent of distributed computing on a global scale.

Stimulus to design innovation from key market #4: US

Whereas R/2 was suited mainly to mainframe computers in larger firms, SAP developed the follow-up product R/3 to run on client-server networks. For both technical and market-related reasons, development of R/3 led to a greater focus on the IT environment of the US. Development of R/3 brought SAP into the realm of UNIX, relational databases, and the programming language C, all developed in the US and central to the later breakthrough of PC-based client/server networks for use by large corporations. Of special importance was the non-proprietary operating system UNIX that could run on multiple hardware platforms. SAP first began to use UNIX as an internal development tool but gradually discovered that, especially in the US, even corporate customers were interested in building UNIX-based IT systems. By 1992 SAP was able to offer R/3 on UNIX machines independently of any hardware manufacturer.

The design of R/3 as a hardware-independent product overcame a significant barrier to global diffusion of SAP software, namely the existence of mutually incompatible mainframe systems. Prior to 1992, SAP's software could be installed only on IBM and Siemens machines. SAP had not originally embarked on developing R/3 for the purpose of overcoming the hardware obstacle; on the contrary, in developing R/3 for server-client architectures, SAP initially targeted sales for the IBM AS/400 series of machines. Hardware independence as a strategic goal emerged only in the course of development (1988–1992). The original product strategy of SAP had been to expand into the market of medium-sized client firms with R/3 while continuing to offer R/2 to larger firms.

However, as the power of ''scalable'' client-server systems increased in the 1990s, R/3-based systems unexpectedly came to provide an alternative to mainframe-based R/2 systems in large firms.[2] As a result, R/3 proved feasible both in comparatively small firms and in extremely large ones, particularly MNCs. In the course of the 1990s, hundreds of multinational companies purchased and implemented R/3 as a tool to integrate their cross-border operations on global client-server IT systems. R/3 became a truly global product, standardized worldwide. In essence, the combination of prior design features and the new client-server architecture of R/3 resulted in a new synthesis of standardization and adaptation in ERP software.[3] In the 1990s, ERP software became a mainstream rather than a specialized product for corporate users. Indeed, the very term ''enterprise resource planning'' was coined in the early 1990s to denote a category of software that lacked a commonly agreed designation (Jacobs and Weston, 2007).

To achieve this, R/3 incorporated additional dimensions of programmability and modularity in design. In simplified terms, the feature of hardware independence in R/3 required programmability, while the feature of scalability

[2] A noteworthy anecdote concerns the key purchase of R/3 in the US by Chevron Oil. The feasibility of deploying R/3 in a large MNC was discovered not by SAP, but by the IT managers at Chevron who had taken a keen interest in UNIX-based systems: ''At first everyone thought that client-server was only for smaller firms. Chevron was the first to break out and understand that client-server was also suited for large companies. Chevron examined, measured, and invested heavily in the R/3 system and finally determined that R/3 had the potential to surpass the mainframe'' (Plattner, 2000, p. 101). ''We won against Oracle, against JD Edwards, SSA, and all others because Chevron wanted a UNIX system. That meant that R/3 was suddenly in the top league, ahead of R/2. We had to alter our complete development ... This is the story of how R/3 ... became a system playing in the US big league of mega-firms with worldwide installations'' (Plattner, 2000, p. 37).

[3] By this time, even in the US some software companies such as JD Edwards and Oracle were beginning to offer increasingly integrated business software products (Jacobs and Weston, 2007). In Europe, Holland-based Baan emerged as another important early vendor of ERP systems able to compete on an international scale. While other major vendors of ERP software had emerged, only SAP was able to offer a client-server version, giving it a head start for the next several years.

(allowing additional servers and clients to be added at will) essentially involved modular design.

The added programmability associated with hardware independence is best explained by the fact that R/3 is written for ''virtual'' computer systems that can be implemented on a combination of different hardware elements. Such a virtual computer system is defined, first, by a set of interfaces (''protocols'' in technical terms) elaborated at the industry level, and second, by basic ''layers'' of software such as operating systems for linking together elements in the client-server network. These protocols and basic software layers redefine the specific hardware devices as virtual devices (in longstanding technical terms, as ''logical devices''). R/3 therefore operates at a higher level of abstraction than the hardware-specific versions of R/2. R/3 is more programmable than R/2 in the sense that it can be adapted to different ''virtual'' computer systems, and hence different hardware configurations. Thus, whereas developing a new version of R/2 for mainframe systems beyond those of IBM and Siemens was prohibitively expensive, the coding of R/3 to run on the Windows NT operating system rather than on UNIX was a simple affair, requiring only five days of coding by a single programmer (Plattner, 2000, p. 42).

The other key feature of R/3 was ''scalability.'' In IT parlance, R/3 came to ''scale higher'' than R/2 as a result of technical advance in chips, PCs, routing technologies, etc. In design terms, scalability involves modularity. Client-server systems are modular; one even refers to ''client modules'' and ''server modules.'' Scalability in software design combines the modular architecture of servers and clients with the capacity to add additional modules (''nodes'') to the network at will, constrained only by the information-processing capacity of what the underlying hardware can handle. While scalability is sometimes considered a property of hardware (since it is the hardware that ultimately determines the upper limits of computing power and feasible system size), it is the modularity and programmability of the underlying software that makes such scalability operational.

Indeed, although modularity is a generic design feature of client-server systems, SAP contributed one very specific modular design innovation to client-server systems that expanded the scalability of such systems. In the design of R/3, SAP pioneered the use of a three-tiered architecture that overcame an important bottleneck in client-server systems. The bottleneck stemmed from the fact that multiple clients need to access the same centralized applications. As the size of the system increases, the same centralized application server has to serve ever more clients, eventually leading to long waiting times or even system crashes. SAP's important modular design remedy was to create an intermediate tier of computing between the decentralized clients and the centralized servers. Previously, client-server architectures were structured in only two tiers, namely client and server (involving the two competing variants of ''Fat-Client'' or ''Fat-Server''). With R/3 SAP introduced the three-tiered system in which multiple application servers (i.e. multiple ''copies'' of the centralized applications) are made available at an intermediate level between the centralized database server and the decentralized client users (Figure 3). R/3 prevents bottlenecks in accessing

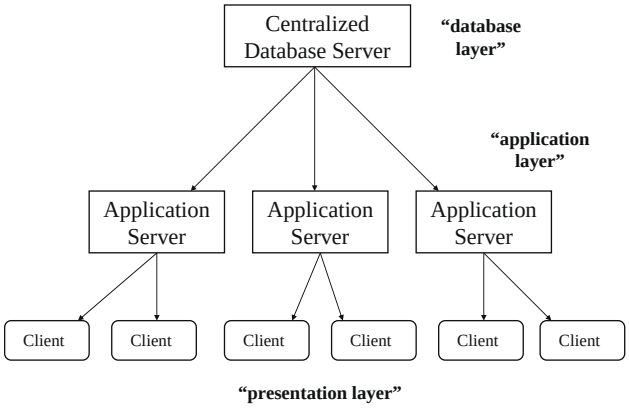

Figure 3 Three-tiered client-server architecture.

applications by modularizing the applications in yet another sense, that of allowing the applications to be duplicated and accessed on multiple physical servers. This additional modular design feature greatly expanded the maximum feasible size of the overall corporate ERP system. The three-tiered client-server architecture developed by SAP has since come to constitute a dominant design in ERP software (Rashid, Hossain, and Patrick, 2002, pp. 7—8).

Discussion: complexity limits to programmability as a means to transcend the standardization-adaptation tradeoff

The SAP case serves both a broad and narrow purpose: more broadly, to illustrate a novel perspective from which to analyze the strategy of MNCs; more narrowly, to shed light on the interplay among design principles. To begin with the broad purpose, the case study shows that in at least some product domains the MNC strategy for coping with the competing imperatives of global standardization and local adaptation of products can be fruitfully analyzed from a product design perspective. Such a perspective contrasts with the two major paradigms for examining MNC strategy, namely the standardization-adaptation paradigm of the international marketing field (involving multivariate tradeoffs on the four dimensions of product, promotion, pricing, and distribution) and the integration-responsiveness paradigm of the international management field (involving mainly organizational decisions about centralization and decentralization of decision-making). This raises two basic questions with respect to the SAP case. First, beyond just the design-specific issues, what are the competitive implications of such a perspective? Second, how generalizable is the SAP case, that is, how relevant might a study of ERP software be to other industries?

Concerning the first issue, the SAP case adds a certain nuance to the well-established principle of market sequencing, that is, the order in which foreign markets are entered. SAP's market sequencing involved first dominating an idiosyncratic home-country market for business software (Germany) that provided a fertile demand environment in which to develop an innovative product (cross-functional business application software) — and probably also shielded the firm from foreign competitors — and then using its gradual internation-

alization to incorporate design enhancements to make the product suitable for a wider range of markets. The case study shows how SAP's design advantage emanated from the mix of market environments in which it participated and the way it was able to combine and reconcile stimuli from heterogeneous market environments in its product design decisions, as summarized in Table 3. In the SAP case, the process happened to be an unplanned one and the sequencing was somewhat serendipitous. SAP's founders have consistently denied ever possessing any grand plan to conquer global markets (Meissner, 1997; Campbell-Kelly, 2003, p. 191). This does not entail, however, that other firms cannot learn from this or that such a process *has* to be unplanned.

Concerning the second question, one can legitimately ask whether the case of ERP software contains the same potential for generalizability as, say, the consumer electronics, telecom equipment, and household products studied by Bartlett and Ghoshal (1989). There is good reason a priori to think that it does. While in the 1980s only large firms with mainframes possessed a complex, modular, multifunctional software suite with millions of lines of code, today every household PC and even many cellphones contain an operating system of comparable complexity designed to enable multifunctionality. Moreover, programmability and modularity are characteristics of not only PCs and cellphones, but also set-top boxes, personal digital assistants, e-book readers, and increasingly even toys and household appliances. The more everyday products come to be made ''intelligent'' and permeated by semiconductors and software, the more relevant the SAP case potentially becomes, although only empirical research can determine the actual degree of relevance.

The SAP case also sheds light on the interplay among design principles, underlining in particular some of the complexity costs associated with the increasing use of programmability to transcend the standardization-adaptation tradeoff. One measure of these costs resides in the thousands of tables that need to be filled out during installation and parameterization of R/2 or R/3. Furthermore, the installation of an ERP system almost invariably entails expensive supplemental software coding to perform customer-specific tasks not covered in the standardized package. In other words, the complexity associated with high levels of parameterization and programmability involves high front-end expenses for the user with a long period of amortization and return (Cooke and Peterson, 1998; Hendricks, Singhal, and Stratman, 2007).[4]

The SAP case disclosed the following basic pattern, as summarized in Figure 4. Increasing levels of programmability in product design involved increasing complexity and implementation costs for the user. One way of reducing these complexity and implementation costs was to incorporate modular design elements. To say that modularity in design helped to palliate the negative side-effects of programmability in design, as implied by Figure 4, is not to say that SAP historically built in modular design features with this specific intention in mind. Nonetheless, the overall *effect* and *interplay* of these design principles was such that one helped reduce the cost of the other for users, that is, modularity in design made high levels of product programmability (and thus, product complexity) more manageable.

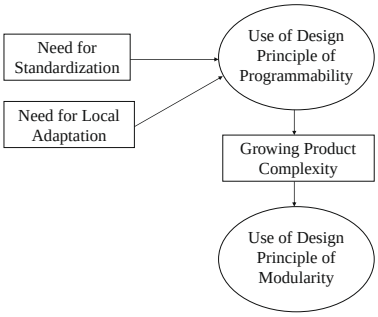

Figure 4 Interplay of programmability and modularity.

As a postscript illustrating the complexity tradeoff that programmability in design entails, SAP actually began to veer away from total product standardization in the late 1990s. Instead of adhering to a completely standardized software suite, SAP introduced ''industry solutions,'' namely supplementary modules to suit the requirements of users in specific industries. Industry solutions were developed for Aerospace, Automotive, Banking, Chemicals, Consumer Products, Engineering, Health care, etc. By introducing industry-specific modules to facilitate ERP implementation and reduce customization costs, SAP took a step ''backward'' from prior efforts to upgrade its ERP software within an ever more universal, one-size-fits-all technology. Following a common trend among ERP providers (Jacobs and Weston, 2007), SAP maintained product standardization across countries but backed off from standardization of its software across industries.

The special relevance of the foregoing case to European firms is obvious. Notwithstanding regulatory harmonization by the European Union, European firms continue to confront considerable diversity in markets on their home continent. The foregoing investigation has identified one little-analyzed mechanism by which the diversity of national markets served by the firm can be converted into a competitive advantage, namely via the design principles of programmability and modularity. In using these principles to improve the terms of the standardization-adaptation tradeoff in product development, SAP transformed its experience in serving heterogeneous markets into design know-how that could be embodied in innovative products suitable for global adoption.

To be sure, exposure to heterogeneity in demand is not always advantageous. The fragmentation of markets has evidently been of little benefit to European producers of standardized software in segments other than ERP; more generally, the conventional wisdom is that European market fragmentation results in handicaps in a number of scale-driven sectors, such as computer hardware and mass entertainment. In other sectors, however, European market diversity is more benign, providing European firms with access to a

[4] One large-scale survey estimated SAP installations in the US to cost an average of $20 million, with larger firms frequently spending over $100 million (Cooke and Peterson, 1998). Of this, only 14% on average was for the software license; 70% of the total cost went toward implementation and 16% toward hardware. This mirrors the ERP rule-of-thumb that the cost of implementing ERP software generally runs above five times the purchase price of the basic software license; the most widespread criticism of R/2 and R/3 (and other ERP systems) voiced in primary research concerns precisely their complexity and difficulty of implementation.

multiplicity of possible solutions to problems (Gambardella and Malerba, 1999; Doz, Santos, and Williamson, 2004).

Finally, the point should be reiterated that programmability and modularity are but two design principles among many that firms can potentially use to resolve standardization-adaptation issues in international product development. Programmability and modularity appear especially well-suited to the requirements of complex IT products like ERP software. For other kinds of products, however, alternative design principles such as ''robustness'' (Rothwell and Gardiner, 1984; Swan et al., 2005) or ''total design'' (Pugh, 1991) may prove more applicable.

Conclusion

Multinational corporations confront the well-known tradeoff between standardization and adaptation in product development. The SAP case presented above exemplified the use of innovation in product design to transcend this tradeoff. Modularity and programmability are not just compromise solutions for coping with these competing imperatives, but rather specific design principles that can be used to reap the advantages of both standardization and adaptation. Although modularity as a design principle has been extensively studied in prior research, the specific relevance of modularity to the standardization-adaptation problem in international product development has been oddly neglected, despite the evident use of modularity in actual MNC practice (Swan et al., 2005). In contrast, the design principle of programmability explored here has received virtually no attention in studies of international product development, although it has been applied in domains like management control systems (Beniger, 1986; Nightingale et al., 2003) and evolutionary theory (Mayr, 1982). A case study design has therefore been considered appropriate both to illustrate the nature of the phenomenon and to anchor the distinction and interplay between modularity and programmability.

Doubts can be raised about the generalizability of the case study in view of the idiosyncratic nature of software. The economics of software entail high costs in development but very low costs in production. This makes it feasible for software providers to include ever greater levels of functionality in their products without increasing production costs significantly. The counter-argument, of course, is that ERP software is simply a manifestation of the information economy which is well-known for such economic characteristics (Shapiro and Varian, 1999). The kind of digital product offered by SAP, while economically idiosyncratic, belongs to a growing proportion of goods in the global high-tech economy. This means that programmability, while arguably applicable as a design principle in only a limited number of product lines, is more likely to be an ascendant than declining approach to manage the standardization-adaptation tradeoff in international product development.

Acknowledgements

The authors thank Oliver Csendes, Phillip Nell, Marcela Miozzo, and three anonymous reviewers for their valuable input and feedback on this article.

References

Baalbaki, I. B. and Malhorta, N. K. (1993) Marketing management bases for international market segmentation: An alternate look at the standardization/customization debate. *International Marketing Review* **10**(1), 19–44.

Baldwin, C. Y. and Clark, K. B. (1997) Managing in an age of modularity. *Harvard Business Review* **75**(September–October), 84–93.

Bartlett, C. A. and Ghoshal, S. (1989) *Managing across borders: The transnational solution*. Harvard Business School Press, Cambridge, MA.

Beniger, J. R. (1986) *The control revolution: Technological and economic origins of the information society*. Harvard University Press, Cambridge, MA.

Campbell-Kelly, M. (2003) *From airline reservations to sonic the hedgehog: A history of the software industry*. MIT Press, Cambridge, MA.

Chesbrough, H. and Teece, D. (1996) When is virtual virtuous? Organizing for innovation. *Harvard Business Review* **74**(1), 65–73.

Cooke, D. P. and Peterson, W. J. (1998) *SAP implementation: Strategies and results*. The Conference Board, New York.

Doz, Y., Santos, J. and Williamson, P. (2004) Diversity: The key to innovation advantage. *European Business Forum* **17**(Spring), 25–27.

Ettlie, J. E. and Subramanian, M. (2004) Changing strategies and tactics for new product development. *Journal of Product Innovation Management* **21**, 95–109.

Gambardella, A. and Malerba, F. (1999) *The organization of economic innovation in Europe*. Cambridge University Press, Cambridge.

Garud, R. and Kumaraswamy, A. (1995) Technological and organizational designs for realizing economies of substitution. *Strategic Management Journal* **16**, 93–109.

Grimshaw, D. and Miozzo, M. (2006) Institutional effects on the IT outsourcing market: Analyzing clients, suppliers and staff transfer in Germany and the UK. *Organization Studies* **27**(9), 1229–1259.

Hall, P. and Soskice, D. (2001) *Varieties of capitalism: The institutional foundations of comparative advantage*. Oxford University Press, Oxford.

Hayes, R. H., Wheelwright, S. C. and Clark, K. B. (1988) *Dynamic manufacturing*. Free Press, New York.

Hendricks, K. B., Singhal, V. R. and Stratman, J. K. (2007) The impact of enterprise systems on corporate performance. A study of ERP, SCM, and CRM system implementations. *Journal of Operations Management* **25**, 65–82.

Jacobs, F. R. and Weston, F. C. T. (2007) Enterprise resource planning (ERP)—A brief history. *Journal of Operations Management* **25**, 357–363.

Jain, S. (1989) Standardization of international marketing strategy: Some research hypotheses. *Journal of Marketing* **53**(1), 70–79.

Kotabe, M. (2003) State-of-the-art review of research in international marketing management. In S. C. Jain (ed.), *Handbook of research in international marketing*. pp. 3–41. Edward Elgar, Cheltenham.

Lane, C. (1989) *Management and labor in Europe: The industrial enterprise in Germany, Britain and France*. Edward Elgar, Aldershot.

Lane, C. and Bachmann, R. (1996) The social constitution of trust: Supplier relations in Britain and Germany. *Organization Studies* **17**(3), 365–395.

Langlois, R. N. (2003) The vanishing hand: The changing dynamics of industrial capitalism. *Industrial and Corporate Change* **12**(2), 351–385.

Langlois, R. N. and Robertson, P. L. (1995) *Firms, markets and economic change: A dynamic theory of business institutions*. Routledge, London.
Lehnerd, A. P. and Meyer, M. H. (1997) *The power of product platforms*. The Free Press, New York.
Lehrer, M. (2000) Has Germany finally fixed its high-tech problem? The recent boom in German technology-based entrepreneurship. *California Management Review* **42**(4), 89–107.
Lehrer, M. (2005) Science-driven vs. market-pioneering high tech: Comparative German technology sectors in the late 19th and late 20th century. *Industrial and Corporate Change* **14**, 251–278.
Lehrer, M. and Asakawa, K. (2002) Offshore knowledge incubation: The ''third path'' for embedding R&D labs in foreign systems of innovation. *Journal of World Business* **37**(4), 297–306.
Levitt, T. (1983) The globalization of markets. *Harvard Business Review* **61**(6), 92–102.
Mayr, E. (1982) *The growth of biological thought*. Belknap Press, Cambridge, MA.
Meissner, G. (1997) *SAP – Die heimliche Software-Macht*. Hoffmann und Campe, Hamburg.
Meyer, M. H. (1998) Product platforms in software development. *Sloan Management Review* **40**(1), 61–74.
Miozzo, M. and Grimshaw, D. (2005) Modularity and innovation in knowledge-intensive business services: IT outsourcing in Germany and the UK. *Research Policy* **34**, 1419–1439.
Miozzo, M. and Grimshaw, D. (2006) *Knowledge intensive business services: Organizational forms and national institutions*. Edward Elgar, Cheltenham.
Mowery, D. C. (1996) *The international computer software industry: A comparative study of industry evolution and structure*. Oxford U. Press, Oxford.
Nightingale, P., Brady, T., Davies, A. and Hall, J. (2003) Capacity utilization revisited: Software, control and the growth of large technical systems. *Industrial and Corporate Change* **12**(3), 477–517.
Nohria, N. and Ghoshal, S. (1997) *The differentiated network: Organizing multinational corporations for value creation*. Jossey-Bass, San Francisco.
Plattner, H. (2000) *Dem Wandel voraus*. Galileo Press, Bonn.
Prahalad, C. K. and Doz, Y. (1987) *The multinational mission: Balancing local demands and global vision*. The Free Press, New York.
Pugh, S. (1991) *Total design: Integrated methods for successful product engineering*. Addison-Wesley, Reading, MA.
Rashid, M. A., Hossain, L. and Patrick, J. D. (2002) The evolution of ERP systems: A historical perspective. In *Enterprise resource planning: Global opportunities and challenges*, (eds) L. Hossain, J. D. Patrick and M. A. Rashid, pp. 1–16. Idea Group Publishing, Hershey.
Rau, P. A. and Preble, J. F. (1987) Standardization of marketing strategy by multinationals. *International Marketing Review* **4**(3), 18–28.
Rothwell, R. and Gardiner, P. (1984) Design and competition in engineering. *Long Range Planning* **19**(2), 78–91.
Samiee, S. and Roth, K. (1992) The influence of global marketing standardization on performance. *Journal of Marketing* **56**(2), 1–17.
Sanchez, R. (1995) Strategic flexibility in product competition. *Strategic Management Journal* **16**(Special Issue), 135–159.
Sanchez, R. and Mahoney, J. T. (1996) Modularity, flexibility, and knowledge management in product and organization design. *Strategic Management Journal* **17**(Special Winter Issue), 63–76.
Shapiro, C. and Varian, H. R. (1999) *Information rules: A strategic guide to the network economy*. Harvard Business School Press, Boston.
Sturgeon, T. J. (2002) Modular production networks: A new American model of industrial organization. *Industrial and Corporate Change* **11**(3), 451–496.
Subramanian, M. and Hewett, K. (1993) Balancing standardization and adaptation for product performance in international markets: Testing the influence of headquarters-subsidiary contact and cooperation. *Management International Review* **44**(2), 171–194.
Swan, K. S., Kotabe, M. and Allred, B. B. (2005) Exploring robust design capabilities, their role in creating global products, and their relationship to firm performance. *Journal of Product Innovation Management* **22**, 144–164.
Takeuchi, H. and Porter, M. (1986) Three roles of international marketing in global strategy: A conceptual framework. In *Competition in global industries* ed. M. Porter, pp. 111–146. Harvard Business School Press, Boston.
Tellis, W. (1997) Application of a case study methodology. *The Qualitative Report* **3**(3)www.nova.edu/ssss/QR.
Theodosiou, M. and Leonidou, L. C. (2003) Standardization versus adaptation of international marketing strategy: An integrative assessment of the empirical research. *International Business Review* **12**, 141–171.
Yin, R. K. (1993) *Applications of case study research*. SAGE, Newbury Park, CA.
Yin, R. K. (2003) *Case study research: Design and methods*. Sage, London.
Zou, S. and Cavusgil, S. T. (2002) The GMS: A broad conceptualization of global marketing strategy and its effect on firm performance. *Journal of Marketing* **66**(4), 40–56.

MARK LEHRER is an Associate Professor of Management at Suffolk University in Boston. After obtaining his PhD at INSEAD, he worked for 2 years at the Wissenschaftszentrum Berlin (WZB) in 1997–1998 and obtained his first professorship at the University of Rhode Island in 1999. Before joining Suffolk University in 2006, he was a guest professor in 2005–2006 in Vienna at the Wirtschaftsuniversität Wien. He has published about 20 articles and book chapters, including in Research Policy, Organization Studies, Journal of World Business, California Management Review, and Industrial and Corporate Change.

MICHAEL BEHNAM is an Associate Professor of Management at Suffolk University in Boston. After obtaining his PhD at the Johann Wolfgang Goethe-University in Frankfurt, he taught at the European Business School in Oestrich-Winkel, Germany, where he completed his postdoctoral thesis. He has published several books and articles in the areas of international strategy, change management, international business ethics, and knowledge transfer.