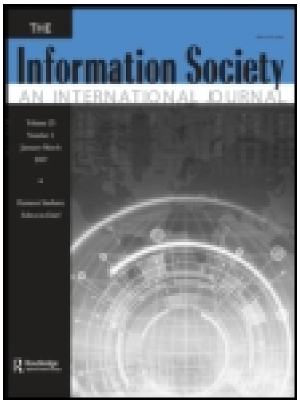

The Information Society: An International Journal
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/utis20

Information Systems and Developing Countries: Failure, Success, and Local Improvisations

Richard Heeks

Published online: 19 Jan 2011.

To cite this article: Richard Heeks (2002) Information Systems and Developing Countries: Failure, Success, and Local Improvisations, The Information Society: An International Journal, 18:2, 101-112, DOI: 10.1080/01972240290075039

To link to this article: http://dx.doi.org/10.1080/01972240290075039

The Information Society, 18:101–112, 2002

0197-2243/02 $12.00 + .00
DOI: 10.1080/0197224029007503 9

Information Systems and Developing Countries: Failure, Success, and Local Improvisations

Richard Heeks

Institute for Development Policy and Management, University of Manchester, Manchester, United Kingdom

This article presents evidence that—alongside the successes—many information systems in developing countries can be categorized as failing either totally or partially. It then develops a new model that seeks to explain the high rates of failure. The model draws on contingency theory in order to advance the notion of design–actuality gaps: the match or mismatch between IS designs and local user actuality. This helps identify two high-risk archetypes that affect IS in developing countries: country context gaps and hard–soft gaps. The model is also of value in explaining the constraints that exist to local IS improvisations in developing countries. Overall, the article shows how model and theory help understand IS cases in developing countries, and equally, how those cases provide valuable data to help develop IS models and theories.

Keywords developing country, information system, evaluation, implementation, failure

DEFINING AND MEASURING SUCCESS AND FAILURE

Do most information systems (IS) projects in developing countries (DCs) succeed or fail? Any attempt to answer this question must start by categorizing success and failure. The basis for categorization was qualitative review of a large number of DC IS case studies (in *Information Technology for Development* and in edited volumes such as Roche & Blaine, 1996; Odedra-Straub, 1996; Avgerou & Walsham, 2000a; IFIP WG9.4, 2000).

Any success/failure categorization runs into some immediate difficulties that this article, while recognizing, cannot completely resolve. The first difficulty is the subjectivity of evaluation—viewed from different perspectives, one person's failure may be another's success (Lyytinen & Hirschheim, 1987; Sauer, 1993). The categorization does try to address this within the limits imposed by the subjectivity of the case study writers themselves.

The second difficulty is the timing of evaluation—today's IS success may be tomorrow's IS failure, and vice versa. Given the reliance on reported cases, which are dominated by cross-sectional rather than longitudinal analyses, this issue could not be adequately incorporated. However, the changing proclivity to failure during IS implementation is discussed.

In analyzing case studies of IS in DCs, three dominant categories of reported outcome emerged, as described later. While not theoretically exhaustive—where, for example, would one place a "failure-followed-by-success" case?—this three-way categorization did cover all the cases reviewed.

First, there was the *total failure* of an initiative never implemented or in which a new system was implemented but immediately abandoned. Such an outcome can be defined relatively objectively. For example, India's Indira Gandhi Conservation Monitoring Centre was intended to be a national information provider based on a set of core environmental information systems. Despite more than a year of planning, analysis and design work, these information systems never became operational, and the whole initiative collapsed shortly afterwards (Puri et al., 2000).

A second possible outcome is the *partial failure* of an initiative, in which major goals are unattained or in which there are significant undesirable outcomes. In some cases, where only a subset of initially stated objectives has been achieved, the notion of partial failure may be relatively straightforward. For example, the Tax Computerization Project in Thailand's Revenue Department set out seven areas of taxation that were to be computerized. At the end of the project, only two areas had been partly computerized, and five others were not operational (Kitiyadisai, 2000).

Received 15 November 2000; accepted 15 July 2001.
Address correspondence to Richard Heeks, Institute for Development Policy and Management, Precinct Center, University of Manchester, Manchester M13 9GH, United Kingdom. E-mail: richard.heeks@man.ac.uk

Where cases are analyzed longitudinally, another type of partial failure can emerge—one that particularly seems to affect developing countries. This is the "sustainability failure" of an initiative that at first succeeds but is then abandoned after a year or so. An example is the creation of a set of touch-screen kiosks for remote rural communities in South Africa's North-West Province. These were initially well received by the communities. However, the kiosks' lack of updated or local content and lack of interactivity led to disuse, and they were removed less than a year later (Benjamin, 2001).

Yet other partial failures are more difficult to identify because identification grapples with the issue of subjectivity. This requires evaluation to ask: "Whose goals are unattained?" and "For whom are the outcomes undesirable?" Answers will only appear where evaluation methods recognize failure's subjectivity, and recognize and interact with multiple stakeholder groups. Such recognition is, unfortunately, rare in evaluations of developing country (and other) IS projects.

There was such recognition in analyzing the Accounts and Personnel Computerization Project of Ghana's Volta River Authority. Most managerial staff in the finance department were pleased with the changes brought by the new system. However, the implementation "bred a feeling of resentment, bitterness and alienation" among some lower-level staff, and led to resistance and nonuse, particularly among older workers (Tettey, 2000, p. 72).

Finally, one may see the *success* of an initiative, in which most stakeholder groups attain their major goals and do not experience significant undesirable outcomes. This, again, requires the relatively sophisticated approach to evaluation that is absent in many cases. In one in-depth evaluation, a South African tire-manufacturing firm introduced a relatively simple workflow tracking system using bar codes on the tires. Analysis from multiple stakeholder perspectives showed that all three key groups—managers, supervisors and workers—perceived the system to have brought benefits to their work (Calitz, 2000).

The Extent of Success and Failure

What proportion of DC IS projects fall into each of the three outcome categories? No one knows for certain. The question is hard enough to answer in the industrialized countries. There, at least, a certain level of surveys, evaluations, and analysis is present (Korac-Boisvert & Kouzmin, 1995; James, 1997; Sauer, 1999; *The Economist*, 2000). On the basis of the range of figures provided in these surveys, one may estimate that something like one-fifth to one-quarter of industrialized-country IS projects fall into the total failure category; something like one-third to three-fifths fall into the partial failure category; and only a minority fall into the success category.

This, at least, can be used as a threshold indicator to answer the question. There is no evidence, nor is there any theoretical rationale, to support the idea that failure rates in developing countries should be any lower than those in industrialized countries. Conversely, there is evidence and there are plenty of practical reasons—such as lack of technical and human infrastructure—to support the idea that failure rates in DCs might be higher, perhaps considerably higher, than this threshold.

What is the evidence relating to IS success and failure in developing countries? Evidence to address the earlier question, and move beyond the threshold estimations just offered, is very limited. In addition to poor recognition of subjectivity and timing of evaluation, the constraints on evidence are several:

- *Lack of literature in general*: Until very recently, the entire literature on IS and developing countries would struggle to fill a single bookshelf. The attention of writers—from researchers to consultants to journalists—has been focused elsewhere.
- *Lack of evaluation*: Those who have the will to evaluate—such as academics—often lack the resources and capacity. Those who have the resources—such as aid donor agencies—often lack the will to evaluate.
- *Focus on case studies*: The literature on IS in DCs has grown, but it is a literature dominated by case studies of individual IS projects. Taken alone, these provide no basis for estimation of overall failure/success rates.

Despite these limitations, there are some glimpses of evidence. An overview of the literature concludes, "successful examples of computerisation can be found . . . but frustrating stories of systems which failed to fulfil their initial promise are more frequent" (Avgerou & Walsham, 2000b, p. 1). This shows up in overall evidence: IT capital shows no significant correlation with productivity in developing countries, while in industrialized countries there is a positive correlation (Kraemer & Dedrick, 2001). Likewise, IT investment shows no significant returns in DCs but 80% gross returns in Organization for Economic Cooperation and Development (OECD) countries.

A few more specific multiple-case studies have been conducted, with examples summarized here:

- Health information systems in South Africa: Braa and Hedberg (2002) reported widespread partial failure of high cost systems with little use of data.
- IS in the Thai public sector: Kitiyadisai (2000) reported "failure cases seem to be the norm in Thailand at all governmental levels."
- Donor-funded IT projects in China: Baark and Heeks (1999) reported that all were found to be partial failures.

- World Bank-funded IT projects in Africa: Moussa and Schware (1992) reported almost all as partial—often sustainability—failures.

Likewise, reports from individual developing countries (e.g., World Bank, 1993; Oyomno, 1996) found IS failure to be the dominant theme.

In summary, the evidence base is not strong—and it urgently needs strengthening—but it all points in one direction: toward high rates of IS failure in developing countries. If this is so, we should seek to understand why. That is the intention of this article—to develop and then apply a new model that helps explain why so many information systems in developing countries fail.

Before moving on to this, though, one further question should be addressed. Is the prevalence of failure a problem? For example, failure can have benefits, especially in relation to learning. Unfortunately, while learning from IS failure does occur, it is generally fortuitous rather than planned (Macias-Chapula, 2000). There are few signs of the presence of learning systems in DC organizations, and some signs of their absence (Shukla, 1997).

In a very direct sense, failure is also a problem because of the opportunity costs of resource investment in failure, as opposed to success. Such opportunity costs are likely to be particularly high in DCs because of the more limited availability of resources such as capital and skilled labor.

Finally, the costs of all types of failure identified here—uncompleted/abandoned projects; projects that fail to meet objectives or which fail to satisfy key stakeholders; and projects that cannot be sustained—are high because only successful projects will ensure global economic convergence (Kenny, 2001). The failures keep developing countries on the wrong side of the digital divide, turning IT into a technology of global inequality. For all these reasons, IS failure is therefore a very real and very practical problem for developing countries that needs to be addressed.

UNDERSTANDING DC IS SUCCESS AND FAILURE

We have an estimation that a significant number of IS projects in developing countries fail in some way. Why should this be? Two bodies of literature relate to this question: the general literature on IS failure (e.g., Lyytinen & Hirschheim, 1987; Horton & Lewis, 1991), and the specific literature on IS failure in developing countries (e.g., Matta & Boutros, 1989; Boon, 1992; Beeharry & Schneider, 1996). Both have been useful in helping build the overall body of knowledge. However, there have been criticisms that such writings have been poor at explaining causes; poor at identifying responses to failure; and that work is too normative and prescriptive, failing to take account of the many differing contexts in which information systems are implemented (Sauer, 1993; Poulymenakou & Holmes, 1996; Montealegre, 1999).

This article therefore seeks to build a new model to understand the success and failure of information systems—a model that will both explain causes and identify responses. In order to avoid the pitfalls of the normative models, the starting point must be contingency, which has a great value in dealing with the complexity of IS and organizations (Robey & Boudreau, 1999). Contingency sees no single blueprint for success and failure in organizational change. Instead, it recognizes that there are situation-specific factors for each information system that will determine success and failure, and hence strategies for success.

Inherent within much of the organizational literature on contingency is the idea of *fit* or *congruence*: of mismatch and match between factors (Lorsch & Morse, 1974; Butler, 1991). Failure is seen to derive from lack of fit between factors; success from congruence between factors. IS implementation models drawing on this theme have involved fit between "tool"/"technology" and "task" (Goodhue & Thompson, 1995; Agarwal & Sinha, 1996), or between multiple factors (Scott Morton, 1991; Southon et al., 1997).

A common base for these models is Leavitt's (1965) diamond, illustrated in Figure 1. In all cases, the concept is one of "dimensional fit": the need for one or more different dimensions of organization and/or environment to be brought into congruence at the same time.

Although valuable, these ideas have two key shortcomings in helping understand and explain IS failure:

- There is a relatively poor conception of organizational change. Earlier models do not adequately account for the fact that, the greater the degree of change, the greater is the risk of failure (Moussa & Schware, 1992; Clemons et al., 1995; Sauer, 1999; Kitiyadisai, 2000). Dimensional fit—requiring different dimensions to be congruent at the same time—is too static a model to handle the process of change. Instead, a more temporal conception of fit is needed, one that conceives the same dimensions being congruent at different times. This

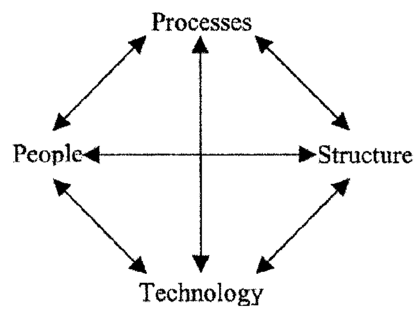

FIG. 1. Dimensional contingency.

requires the notion of fit to be altered: It moves from the dimensional fit models' rather vague notion of "internal consistency/congruence" (since each dimension to be matched is different from all other dimensions) to the very specific notion of "match" between the same dimensions (Venkatraman, 1989). However, if dimensions exactly match at different times, then there is no organizational change. Therefore, exact match on all dimensions is not the desired outcome; intentional difference is the outcome. One may simply say that—as earlier—the greater the mismatch (i.e., the greater the change), the greater is the risk of failure, and the greater the match (i.e., the smaller the change), the greater is the likelihood of success.

- There is a relatively poor conception of technology. Earlier contingency models conceive technology to be a black box artifact that is separable from other contextual dimensions (Sauer, 1999). Yet this is clearly not so. As well as the physical artifact, technologies contain within them an inscribed "vision of (or prediction about) the world" (Akrich, 1992, p. 208). This "world-in-miniature" includes inscriptions of how processes will be undertaken; of the values that people will have; of the structures in which they are to be placed; and so on. Technology must therefore be seen not in a reductionist manner as a separate dimension, but in a systemic manner as a group of related dimensions.

Putting together these two ideas—temporal fit, and a systemic view of technology—we arrive at the interim model summarized in Figure 2. This regards the match or mismatch between a system now and that system in the future as an important determinant of the likelihood of the system in the future falling into one of the success or failure categories identified. Further, it regards greater match as linked with greater likelihood of success, and greater mismatch as linked with greater likelihood of failure. Further still, greater mismatch would be linked with greater likelihood of total, as opposed to partial failure. To avoid constant repetition, however, both categories are bundled together as "failure" in the discussion that follows.

Design–Actuality Gaps

How could this interim model of temporal, systemic fit be operationalized and tested? We need to simultaneously evaluate the current system and the future system. Yet, by definition, they cannot simultaneously exist. It is relatively easy to assess the current "actuality" in a location. But in order to assess the future, we must assess instead the representation of an intended future—an intended future that is represented in a design for the system. The model to be used here is therefore based on an assessment of the match or mismatch between local actuality ("where we are now") and system design ("where the design wants to get us"). Put simply, we refer to this as the *design–actuality gap*.

In practice, because of subjective expectations about the future and subjective perceptions of reality, it could be argued that every individual IS stakeholder has their own design and their own version of actuality. Among these myriad design–actuality gaps, we must necessarily simplify the model. Drawing on another thread within the failure literature (Lyytinen & Hirschheim, 1987; Sauer, 1999), the two key homogenized stakeholders in the model presented here will be the designers who create the dominant IS design, and the users who populate the local actuality.

These groups are especially valuable to an understanding of failure given their dislocation, in both psychological and even physical terms, as part of the IS implementation process. However, this simplification does impose limits—for example, limiting subjective partial failures to a consideration of the objectives of these two stakeholder groups alone.

What could be relevant dimensions of this design–actuality gap between the designers' dominant design and the

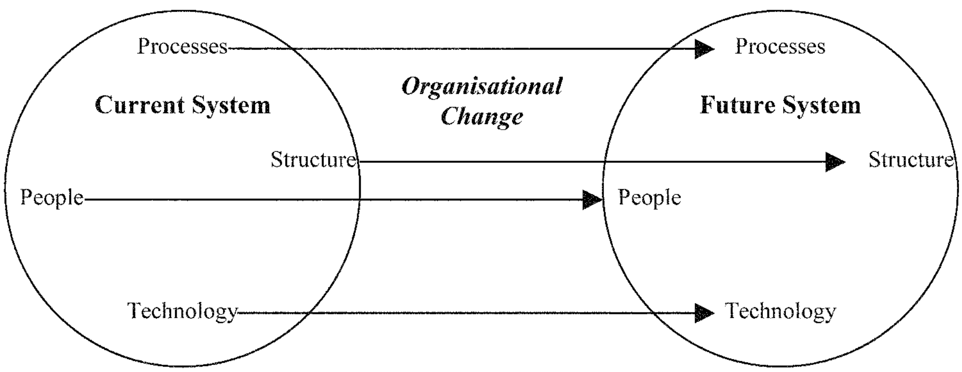

FIG. 2. Temporal, systemic contingency.

local actuality of the users? The dimensions could be built up in a number of ways: theoretically on the basis of information systems literature; descriptively on the basis of a straightforward delineation of components of an information system; and analytically on the basis of case studies. An amalgam of all three approaches is attempted here.

In all, as noted earlier, the design is a representation of an intentional future. It is a world-in-miniature that contains elements that have been inscribed either explicitly or implicitly. These elements include:

- *Components from the designers' own context*: IS design is a situated action—an action "taken in the context of particular, concrete circumstances" (Suchman, 1987, p. viii). This action draws elements of that context into the design: "Our technologies mirror our societies. They reproduce and embody the complex interplay of professional, technical, economic and political factors" (Bijker & Law, 1992, p. 3). Designers themselves are part of and shaped by that context, so their own cultural values, objectives, etc. will be found inscribed in the design (Shields & Servaes, 1989; Braa & Hedberg, 2002).
- *Conceived assumptions about the situation of the user*: This includes assumptions about the users' activities, skills, culture, and objectives, and assumptions about the user organization's structure, infrastructure, etc. (Boehm, 1981; Suchman, 1987; Clemons et al., 1995; Wynn & deLyra, 2000).

Combined with more descriptive material on information systems, these theoretical ideas build to create seven dimensions of relevance to design–actuality gaps: information (data stores, data flows, etc.); technology (both hardware and software); processes (the activities of users and others); objectives and values (the key dimension, through which factors such as culture and politics are manifest); staffing and skills (both the quantitative and qualitative aspects of competencies); management systems and structures; and other resources (particularly time and money).

Analysis of IS failure and success case studies, from developing countries and beyond, shows that these seven dimensions provide a model that can be applied in practice to a wide range of case studies (Heeks, 2001). Examples of such case study analysis are given later in the article, but the model can be summarized as shown in Figure 3. For each of the seven dimensions, the gap between design and actuality can be assessed and rated (e.g. low, medium, high). Overall ratings will give a sense of mismatch between design and actuality and, hence, a view of the likelihood of failure.

EXPLAINING DEVELOPING COUNTRY INFORMATION SYSTEM FAILURE

Robey and Boudreau (1999, p. 181) stated that "information technologies are produced by the very social structures that they promise to transform." This is not so because the context of design/production is not the same as the context of use. In studying industrialized country IS cases, however, it can be hard to disentangle these two because

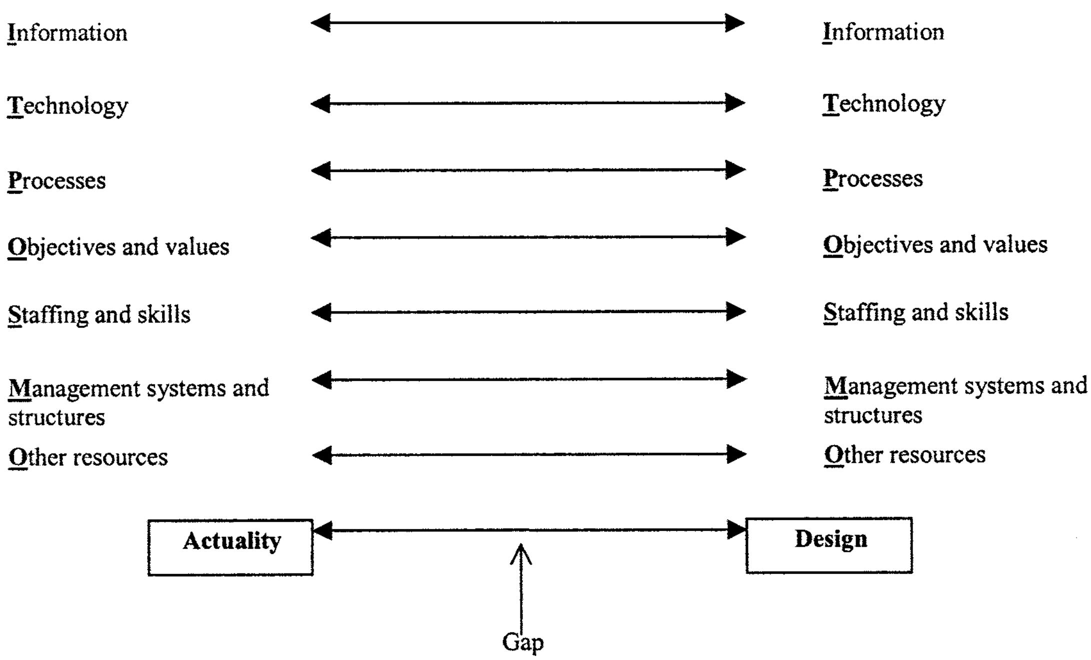

FIG. 3. Design–actuality gaps.

of the proximity (in all senses) and similarity of designer and user context. Designers in industrialized countries may still find themselves "automating a fiction" (Bannon, 1998, p. 58), but design–actuality differences can be subtle, implicit, and hard to identify. It can therefore be hard to think beyond the black box.

Herein lies the value of working with IS cases from developing countries. The contexts of designer and user are often distant in physical, cultural, economic, and many other ways. The remoteness of designers means that their contextual inscriptions are liable to be significantly different from user actuality. So, too, are the inscribed assumptions that remote designers make about that actuality. Design–actuality gaps are therefore more extreme and more explicit and, as a result, are easier to identify and to understand. DC cases therefore provide valuable data that helps illuminate both IS failure and underlying structures and processes. Put another way, DC cases make it easier to move beyond the black box (Akrich, 1992).

The remoteness of designs and of dominant design stakeholders can happen in a number of ways, but the domain of developing country information systems is particularly dominated by the transfer of industrialized country designs to DC actualities (Barrett et al., 2001). This domination comes partly from the economics of innovation and the domination of IT/IS-related research and development (R&D) systems by industrialized country companies and researchers. It comes partly from the economics of business, which sees industrialized-country organizations able to invest more and earlier in new information systems than their DC counterparts. It comes partly from the economics and politics of aid, which has been dominated by a flow of resources and artifacts from industrialized to developing countries rather than, for instance, between DCs. It even comes partly from cultural attitudes in DCs, where belief in the superiority of imported items is sometimes strong (Heeks, 1996). Finally, the whole process has been both strengthened and enabled by globalization: an activity that has, in the main, carried ideas and systems from industrialized to developing countries.

The most extreme form occurs when industrialized-country designers create an information system within and for an industrialized-country context, and that IS is subsequently transferred to a developing country. In such situations, the actuality of local conditions in the developing country will not have been considered at all in the original design, and a considerable design–actuality gap is therefore likely, leading to a significant risk of IS failure.

An example can be drawn from the Philippines. There, an aid-funded project to introduce a field health information system was designed according to an American model that assumed the presence of skilled programmers, skilled project managers, a sound technological infrastructure, and a need for information outputs like those used in an American health-care organization (Jayasuriya, 1995). In reality, none of these was present in the Philippine context. The result was a large design–actuality gap along dimensions including information, technology, and staffing/skills. The outcome was that the information system failed.

Even if some effort is made to develop an information system specifically for a developing-country organization, similar problems can arise. Industrialized-country stakeholders, such as consultants or IT vendors or aid donors, often dominate the IS design process in DCs. Those stakeholders bring their context with them and, even if located in a developing country, they will inscribe that context into their IS designs: inscriptions that will mismatch DC actuality. Some stakeholders bring with them the "If it works for us, it'll work for you" mentality that makes no attempt to differentiate between industrialized and developing contexts. Others will differentiate, but—given their poor understanding of local DC conditions—their assumptions about user actuality will be incorrect. In all cases, large design–actuality gaps and high failure risks are the outcome.

Problems can even occur where stakeholders from industrialized countries are not directly involved. One could argue that "the West" (as shorthand for industrialized countries) is not just a physical location; it is also a state of mind that has now come to exist for increasing numbers of key figures in developing-country organizations. This transfer of context could be said to occur directly through education of these key figures in the West or even in Western-developed educational systems, and indirectly through the leverage gained by Western domination of economic, political, and cultural resources and channels. These individuals therefore might be seen to act as Trojan horses. Having been indoctrinated into a industrialized-country mindset, they then devise Western-inspired designs within developing-country organizations.

Of course, such an example is an indication that great care must be taken with these notions of industrialized and developing country. They are not neat dichotomous categories, but represent extremes on a continuum. Country context gaps will be easiest to identify and comprehend at the extremes, such as a package designed by and for rich Americans being implemented in a poor Ugandan community. Context gaps will become fuzzier as both the context of design and the context of use approach the center of the continuum.

In all, though, these country context gaps do provide an overview of differences between industrialized and developing countries: differences that lead to design–actuality gaps and risk of IS failure. However, as discussed next, there is an additional hard–soft gap often overlain on the country context gaps which increases that risk.

Hard–Soft Gaps

Information systems for development have been affected by the intimate three-way association of information technology, modernization, and Western rationalism (Shields & Servaes, 1989; Avgerou, 2000; Tettey, 2000).

Information systems per se have a tendency to be designed according to models of rationality. In part, this occurs because of the continuing emphasis on IT within information systems change. Many IT designers tend to draw from and work within a rationalist tradition (Mundy et al., 2001). More generally, technology is conceived as an objective and rational entity, not as something (as described earlier) that incorporates particular political and cultural values. The tendency toward rationality in IS design is reinforced by the rationality of the modernization agenda that carries innovations from industrialized to developing countries. It is also reinforced by the "discourse of rationality"—the way in which users and others feel it is only legitimate to discuss organizational issues in rational terms, suppressing more behavioral, political discourse (Heeks, forthcoming).

This combination can readily be seen at work in the agendas of many donor agencies. For them, the overall purpose of development is the creation of economic rationalism within DC economic systems. IT is seen as a key tool in achieving this, and becomes part of a technically rational and technologically determinist agenda that focuses on the digital divide, on "eDevelopment," and on IT infrastructure (Wilson & Heeks, 2000). Any IT problems are, in turn, seen as best solved by resort to market rationality.

One could argue the validity of rational models in an industrialized country context. Here, though, the problem is the gap that can sometimes exist between the rationality of IS design and the political/behavioral actualities of developing-country organizations. These latter realities have been extensively described, and gaps between "hard" rational design and "soft" political actualities are summarized in Table 1 (Heeks & Mundy, 2001).

Failure is frequently the outcome of such gaps between hard design and soft actuality. For instance, geographic information systems (GIS) are seen to incorporate a number of assumptions and requirements that derive from Western rationalism (Walsham, 2000). Introduction of GIS in developing countries has therefore been problematic.

This was the case when a GIS was introduced by the Indian Ministry of Environment and Forests for forestry management. As analysed by Barrett et al. (2001), identified differences can be related to dimensions in Table 1 that include information, technology, processes, objectives/values, and staffing/skills:

- The GIS design assumed reliance on formal types of information borne via technical channels "as compared to the informal channels of information" that were used in practice (p. 18).
- The GIS design assumed "a form of working culture wherein decisions are made on criteria of rationality and principles of cartographic science." (p. 14). This mismatched an actuality of politicized decision making.
- The GIS design representations of the forest conflicted with the actuality of forest officers'

TABLE 1
Differences between hard and soft models

Dimension	"Hard" rational design	"Soft" political actuality
Information	Emphasis on standardized, formal, quantitative information	Emphasis on contingent, informal, qualitative information
Technology	A simple enabling mechanism	A complex, value-laden entity: status symbol for some, tool of oppression for others
Processes	Stable, straightforward, and formal; decision outcomes as optimal solutions based on logical criteria	Flexible, complex, constrained, and often informal; decision outcomes as compromises based on "power games"
Objectives and values	Formal organizational objectives	Multiple, informal, personal objectives
Staffing and skills	Staff viewed as rational beings	Staff viewed as political beings
Management systems and structures	Emphasis on formal, objective processes and structures	Emphasis on informal, subjective processes and structures
Other resources: time and money	Used to achieve organizational objectives	Used to achieve personal objectives

representations, which did not see "land as something that is out there and that can be objectively measured and standardized in GIS models" (p. 13).
- The GIS design required values of trust in the technology, in "new forms of rationality" (p. 19), and in persons unknown and absent. This mismatched the real values of trust in persons known and present.

The result was a significant design–actuality gap along several of the Table 1 dimensions, and the outcome was failure: "there were no real operational systems established by the end of the project period" (p. 10).

THE ROLE OF LOCAL IMPROVISATION

In some ways, the archetypes of country context and hard–soft failure just presented take a rather static view—a cross-sectional perspective of large mismatches between design and user actuality that mean a high risk of failure. The limitations of a cross-sectional view of failure were noted earlier, and likewise, neither design nor actuality should be seen as static; they are dynamic. They both change constantly throughout all phases of an IS project, as, therefore, do design–actuality gaps.

Gaps may increase during implementation and operation. For example, sustainability failures frequently occur when design and actuality spring apart. Typical examples from DC IS cases include actuality changes:

- Along the *other resources* dimension when donor funds are withdrawn.
- Along the *staffing and skills* dimension when key IS staff quit.
- Along the *objectives and values* dimension when senior-level champions move on.

Alternatively, it may simply prove impossible—as with the North-West Province case cited earlier—to bring design and actuality together.

If, on the other hand, the success rate of DC information systems projects is to increase, design–actuality gaps need to be reduced or even closed. This means:

- *Actuality improvisation*: changing local actuality to make it closer to IS design.
- *Design improvisation*: changing the (often "imported") IS design to make it closer to DC user actuality.

Such changes may often be local improvisations: actions by local stakeholders who are not so remote from the context of IS implementation and use. In conceptual terms, this represents a shift from the sense of planned change associated with the country context and hard–soft archetypes to a sense of emergent change that is associated with local improvisations. These emergent changes can be envisaged as continuous reciprocating improvisations between design and actuality that, if success is to be achieved, will seek accommodation and adaptation between design and actuality sufficient to achieve workable closure (Orlikowski, 1996).

In industrialized-country IS cases, local improvisations can often be quite subtle and hard to identify. In DC cases, the improvisations may well have greater clarity because of the greater initial gap between design and actuality. For example, in relation to *processes* and *management systems* dimensions, an original design option for a new hospital IS in Guatemala was to reengineer administrative processes to make them more efficient (Silva et al., 2000). But, in reality, hospital directors supported current procedures and wanted controls to remain in place to ensure corruption was held in check. The design was therefore amended to ensure that these current work processes were supported by the new system.

This was a design improvisation. An example of an actuality improvisation occurred during the introduction of MIS into private-sector enterprises in Sri Lanka (Goonatilake et al., 2000). Here the rational design of the MIS often mismatched the rather chaotic nature of most enterprise procedures. Actuality was altered by, prior to computerization, ensuring the introduction of basic manual production planning, control and accounting procedures. Computerization could subsequently proceed with a greater chance of success.

THE CONTEXT FOR LOCAL IMPROVISATION

Local improvisations are situated actions affected by and affecting the context of their execution. To what extent does that context support or constrain improvisation? Cooper (2000) identified a whole raft of factors that affect creative improvisations during the IS life cycle. Here we focus on just a few that help explain why gap-reducing improvisation is more difficult to achieve in developing countries, and hence why risks of failure are higher than in industrialized countries.

Technology

The IS application itself will significantly affect improvisation. We can delineate a continuum of such applications based on Akrich's (1992) notions of obduracy and plasticity of artifacts (see Figure 4).

At one end of the continuum, *design-imposing applications* can be seen as largely constraining. These are applications that contain "deep inscriptions" and that—to be successful—either require or impose a strong set of processes, values, competences, systems, etc. An example

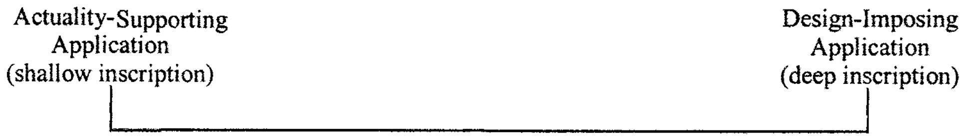

FIG. 4. Deep- versus shallow-inscribed applications.

would be a decision support system. This requires or imposes a series of rational design inscriptions: about the objectivity of information that is present in the system; about the formality of processes and management involved; about the skills and role of people; about the presence of organizational strategies; about the rationality of organizational culture; about the absence of organizational politics; and so on.

This deep design inscription will often create a large initial design–actuality gap. At the same time, though, deep design inscription severely constrains improvisation. It leaves little room for local design improvisation and it predetermines the direction and nature of any local actuality improvisations. IS failure would be a characteristic outcome.

At the other end of the continuum, *actuality-supporting applications* can be seen as more enabling. These are applications that contain relatively "shallow inscriptions;" that is, they require or impose few systemic components. An example would be a word-processing application. This makes some design assumptions about skills, about technical infrastructure, and about cultural values related to technology and to documentation. However, these assumptions are far fewer than for the decision support system.

Shallow design inscriptions will be associated with a smaller initial design–actuality gap. They will also permit greater local improvisations to the design of systems that use the application and they will do relatively little to constrain local actuality improvisations. As a result, such applications have succeeded far more often than design-imposing applications (Heeks et al., 2000).

Ceteris paribus, actuality-supporting applications will therefore be a more appropriate choice for developing countries. However, as Akrich (1992) described, it is the design-imposing approach that has been typical when industrialized-country designers design technologies for developing countries. In such cases, it may be the deliberate intention of those designs to choke off opportunities for local improvisation. But this therefore increases the risk of failure.

Nature of Design

Local design improvisation will be affected by the balance between explicit and implicit components within the design. Explicit components—such as the total cost or the type of computers to be used—will be relatively easy to alter and to bring closer to user actuality. Implicit components—such as designer assumptions about the values or knowledge of local users—will be much harder to alter. Unfortunately, the process of technology transfer to developing countries tends to increase the extent of implicit inscription, therefore reducing local room for maneuver.

Local room for maneuver is also dependent on design divisibility—the extent to which the design comes as a series of divisible subelements. High divisibility increases the opportunities for successful local improvisation. It reduces barriers to improvisation, increases opportunities for learning, and allows improvisations that reduce design–actuality gaps by limiting the extent of change during any given time period.

The Volta River Authority information system cited earlier (Tettey, 2000) avoided the total failure category because its design had two divisible characteristics (see Figure 5):

- Modularity (supporting one business function at a time by allowing separation of, for example, accounting and personnel functions).
- Incrementalism (providing stepped levels of support for business functions by allowing separation of, for example, clerical and management support).

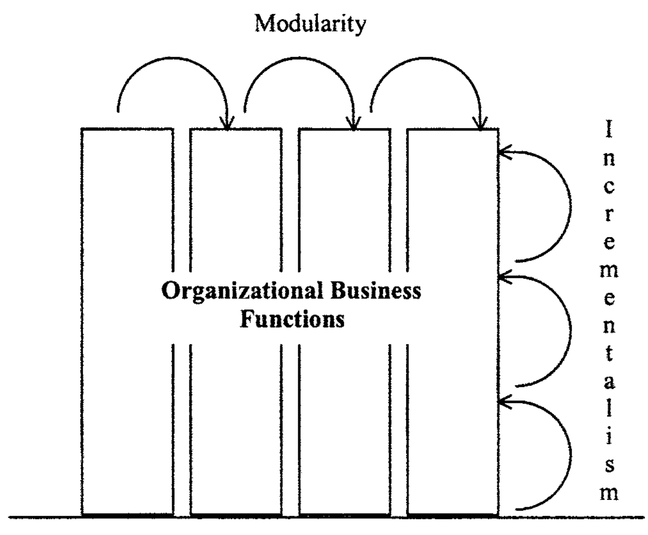

FIG. 5. Modularity and incrementalism in IS projects.

This design divisibility meant staff could learn from early, relatively small failures, and could address subsequent improvisations of both design and actuality to manageable project components. They were not overwhelmed as they would have been by a single, whole system design.

Design divisibility is therefore a frequently cited prophylactic against failure that should be adopted more widely. However, many donor-funded IS projects in developing countries take the opposite approach, partly because of short donor time scales and attention spans (Cain, 2001). Where design comes as this single whole, "big bang" implementation, opportunities for local improvisation are reduced and risks of failure correspondingly increase.

Local Capacities

Technology and design factors may allow room for local improvisation, but the ability of implementers in developing countries to enact such improvisations will depend partly on local capacities. A wide range of such local capacities is required, but there is a central requirement for hybrids (Earl, 1989). Hybrids understand both context, organization, and work processes of their sector *and* the role of information systems, as illustrated in Figure 6 (Heeks, forthcoming). As such, they can bridge the contexts and assumptions of both technical designer and business-oriented user. They can therefore play a key role in the improvisation of both design and actuality, and help to improve success rates. Yet, to date, schemes to develop hybrids in developing countries and, hence, DC hybrids themselves have been virtually nonexistent, thus hampering improvisation (Mundy et al., 2001).

More generally, those who argue for a greater emphasis on contingent improvisation, and reduced emphasis on initial design and planned change, are basing their ideas on an "expert model" that assumes the presence of confident, competent local improvisation capacity (Suchman, 1987; Orlikowski, 1996). In many developing-country contexts, IS implementation may be more strongly associated with a "novice model" where such attitudes and skills are largely absent. In such situations, there will necessarily have to be less emphasis on emergent improvisation and greater emphasis on the initial design, which is likely to increase risks of IS failure.

Improvisation-Supporting Approaches and Techniques

The literature on creativity and improvisation indicates that they can be supported by a variety of other contextual elements and techniques. One such is participative approaches to implementation, using techniques involving group working and end-user involvement (Cooper, 2000). These techniques seek to bridge the contextual gap between design and use.

Such approaches and techniques are sometimes recommended by some of the more analytical literature on IS in DCs. However, great care must be taken here. Such techniques have normally been developed in and for industrialized-country organizations. The points made earlier—about design inscription from industrialized-country contexts and potential mismatch with DC contexts—will therefore apply.

Thus, in practice, participative IS techniques were a failure in Mexico's General Hospital (Macias-Chapula, 2000). Likewise, an end-user development initiative for health IS in South Africa was "an abysmal failure" (Braa & Hedberg, 2002). These implementation techniques failed because there was too large a gap between the design assumptions and requirements of those techniques and the actuality of organizations into which they were introduced. Some improvisation-supporting techniques may therefore be hard to implement in DC contexts, adding one further difficulty to developing-country attempts to address IS failure through local improvisation.

Such findings are also a reminder that we should extend use of the design–actuality gap model presented in this article. It can be used to assess not just the feasibility of a particular information system design (the content/"what" of information systems), but also the feasibility of particular IS implementation techniques (the process/"how" of information systems). As such, it represents a powerful management tool for those involved in the development of information systems in developing countries.

CONCLUSIONS

The concept of fit and of design–actuality gaps can be seen—implicitly—to underlie a number of the more

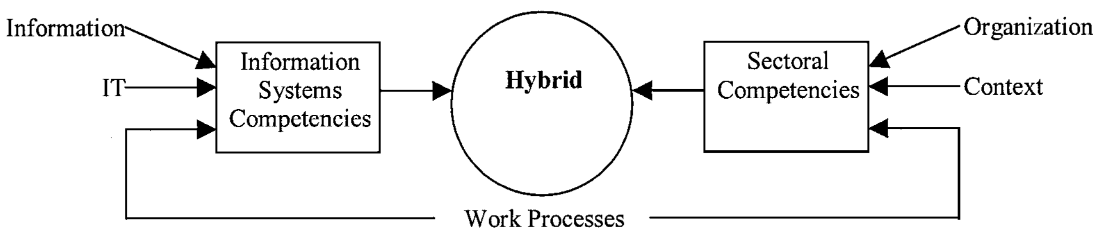

FIG. 6. The competencies of hybrids.

analytical writings on information systems (Clemons et al., 1995; Robey & Boudreau, 1999), on technology transfer (e.g., Akrich, 1992), and on IS in developing countries (e.g., Silva et al., 2000; Barrett et al., 2001). In this article, these implicit concepts have been turned into an explicit model.

That model and its underlying theories have a two-way relationship with information systems in developing countries. First, as has been shown within this article, model and theory serve DC IS cases. The model has analytical strength in helping to understand IS failure in developing countries: both problematic archetypes of failure, and barriers to the local improvisations that attempt to avert failure. It can also claim a predictive strength of potential pre hoc identification of likely failures, and a practical strength of recommendations for action (Heeks, 2001).

But what this article also demonstrates is the converse—the way in which information systems cases in developing countries help to elaborate such models and, in so doing, contribute to underlying theory. At one level, for example, DC IS cases are informing IS failure theory. They do so in quantitative terms because—to developing countries' cost—there is a prevalence and accessibility of IS failures not so readily found in industrialized countries. In qualitative terms, these cases provide a clarity of contingencies and dimensional differences not often found in industrialized-country cases.

At a deeper level, the ideas discussed here are informed by ideas on situated action and by the interrelationship between context and action. IS cases from industrialized countries can be challenging to interpret since the cycles of reproduction and amendment take place within the same context. Adding in the "stretch" of technology transfer throws new light onto the theory by pulling the cycles of interaction between context and action into a spiral that moves between different contexts. This stretch across place and time allows a clearer differentiation between the initial context, the design process, the design, the implementation context, the implementation process, and the ongoing contextual changes. IS cases from developing countries therefore provide fertile ground for helping understand the complex interplay of action and context that underlies all organizational change.

REFERENCES

Agarwal, R., and Sinha, A. P. 1996. Cognitive fit in requirements modeling: A study of object and process methodologies. *Journal of Management Information Systems* 13(2):137–163.

Akrich, M., 1992. The de-scription of technical objects. In *Shaping technology/building society*, eds. W. E. Bijker and J. Law. Cambridge, MA: MIT Press.

Avgerou, C. 2000. The multiple rationalities of information systems development. In *Information flows, local improvisations and work practices*. Proceedings of the IFIP WG9.4 Conference 2000. Cape Town: IFIP.

Avgerou, C., and Walsham, G. eds. 2000a. *Information technology in context*. Aldershot, UK: Ashgate.

Avgerou, C., and Walsham, G. 2000b. IT in developing countries. In *Information technology in context*, eds. C. Avgerou and G. Walsham, pp. 1–8. Aldershot, UK: Ashgate.

Baark, E., and Heeks, R. 1999. Donor-funded information technology transfer projects. *Information Technology for Development* 8(4):185–197.

Bannon, L. 1998. Computer-supported collaborative working: Challenging perspectives on work and technology. In *Information technology and organizational transformation*, eds. R. D. Galliers and W. R. J. Baets, pp. 37–64. Chichester, UK: John Wiley.

Barrett, M., Sahay, S., and Walsham, G. 2001. Information technology and social transformation. *The Information Society* 17(1):5–20.

Beeharry, A., and Schneider, G. M. 1996. Creating a campus network culture in a newly developing economy. *Information Technology for Development* 7(1):3–16.

Benjamin, P. 2001. Community development and democratisation through information technology. In *Reinventing government in the information age*, ed. R. Heeks, pp. 194–210. London: Routledge.

Bijker, W. E., and Law, J. 1992. General introduction. In *Shaping technology/building society*, eds. W. E. Bijker and J. Law, pp. 1–14. Cambridge, MA: MIT Press.

Boehm, B. 1981. *Software engineering economics*. New York: Prentice Hall.

Boon, J. A. 1992. Information and development. *The Information Society* 8(4):227–241.

Braa, J., and Hedberg, C. 2002. Developing district-based health care information systems. *The Information Society* 18(2):113–127.

Butler, R. 1991. *Designing organizations*. New York: Routledge.

Cain, P. 2001. Automating personnel records for improved management of human resources. In *Reinventing government in the information age*, ed. R. Heeks, pp. 135–155. London: Routledge.

Calitz, A. P. 2000. Transforming work practices with IT. In *Information flows, local improvisations and work practices*. Proceedings of the IFIP WG9.4 Conference 2000. Cape Town: IFIP.

Clemons, E. K., Thatcher, M. E., and Row, M. C. 1995. Identifying sources of reengineering failures: A study of the behavioral factors contributing to reengineering risks. *Journal of Management Information Systems* 12(2):9–36.

Cooper, R. B. 2000. Information technology development creativity: A case study of attempted radical change. *MIS Quarterly* 24(2):245–276.

Earl, M. J. 1989. *Management strategies for information technology*. Hemel Hempstead, UK: Prentice Hall.

The Economist. 2000. No gain without pain, Government and the Internet Survey. 24 June:7–10.

Goodhue, D. L., and Thompson, R. L. 1995. Task-technology fit and individual performance. *MIS Quarterly* 19(2):213–236.

Goonatilake, L., Maizza-Neto, O., and Jayawardene, P. 2000. Enhancing enterprise competitiveness in developing countries through the promotion of management information and benchmarking tools. In *Information flows, local improvisations and work practices*. Proceedings of the IFIP WG9.4 Conference 2000. Cape Town: IFIP.

Heeks, R. 1996. *India's software industry*. New Delhi: Sage.

Heeks, R., ed. 2001. *Reinventing government in the information age*. London: Routledge.

Heeks, R. Forthcoming. *e-Government*. London: Sage.

Heeks, R., and Mundy, D. 2001. Information systems and public sector reform in the Third World. In *The internationalization of public management*. eds. W. McCourt and M. Minogue, pp. 196–219. Cheltenham, UK: Edward Elgar.

Heeks, R., Mundy, D., and Salazar, A. 2000. Understanding success and failure of healthcare information systems. In *Healthcare information systems*, ed. A. Armoni, pp. 95–128. Hershey, PA: Idea Group.

Horton, F. W., and Lewis, D., eds. 1991. *Great information disasters*. London: ASLIB.

IFIP WG9.4. 2000. *Information flows, local improvisations and work practices*. Proceedings of the IFIP WG9.4 Conference 2000. Cape Town: IFIP.

James, G. 1997. IT fiascoes . . . and how to avoid them. *Datamation*. November:84–88.

Jayasuriya, R. 1995. Health care informatics from theory to practice. In *Medinfo '95*, eds. R. A. Greenes, H. E. Peterson, and D. J. Protti, pp. 1603–1607. Edmonton: Healthcare Computing and Communications Canada.

Kenny, C. 2001. *The Internet and economic growth in LDCs*. Draft paper. Washington, DC: World Bank.

Kitiyadisai, K. 2000. The implementation of IT in reengineering the Thai Revenue Department. In *Information flows, local improvisations and work practices*. Proceedings of the IFIP WG9.4 Conference 2000. Cape Town: IFIP.

Korac-Boisvert, N., and Kouzmin, A. 1995. Transcending soft-core IT disasters in public sector organizations. *Information Infrastructure and Policy* 4(2):131–61.

Kraemer, K., and Dedrick, J. 2001. *The productivity paradox: Is it resolved? Is there a new one? What does it all mean for managers?* Irvine, CA: Center for Research on Information Technology and Organizations, University of California.

Leavitt, H. J. 1965. Applying organizational change in industry: Structural, technological and humanistic approaches. In *Handbook of organizations*, ed. J. G. March, pp. 1144–1170. Chicago: Rand McNally.

Lorsch, J. W., and Morse, J. J. 1974. *Organizations and their members: A contingency approach*. London: Harper and Row.

Lyytinen, K., and Hirschheim, R. 1987. Information systems failures: A survey and classification of the empirical literature. *Oxford Surveys in Information Technology* 4:257–309.

Macias-Chapula, C. A. 2000. Issues learned, challenges and opportunities to implement a library automation project at Mexico's General Hospital. In *Information flows, local improvisations and work practices*. Proceedings of the IFIP WG9.4 Conference 2000. Cape Town: IFIP.

Matta, K. F., and Boutros, N. E. 1989. Barriers to electronic mail systems in developing countries. *The Information Society* 6(1–2): 59–68.

Montealegre, R. 1999. A case for more case study research in the implementation of information technology in less-developed countries. *Information Technology for Development* 8(4):199–207.

Moussa, A., and Schware, R. 1992. Informatics in Africa. *World Development* 20(12):1737–1752.

Mundy, D., Kanjo, C., and Mtema, P. 2001. Meeting training needs for information age reform. In *Reinventing government in the information age*, ed. R. Heeks, pp. 271–292. London: Routledge.

Odedra-Straub, M., ed. 1996. *Global information technology and socio-economic development*. Nashua, NH: Ivy League.

Orlikowski, W. J. 1996. Improvising organizational transformation over time: A situated change perspective. *Information Systems Research* 7(1):63–92.

Oyomno, G. Z. 1996. Sustainability of governmental use of microcomputer-based information technology in Kenya. In *Global information technology and socio-economic development*, ed. M. Odedra-Straub, pp. 19–34. Nashua, NH: Ivy League.

Poulymenakou, A., and Holmes, A. 1996. A contingency framework for the investigation of information systems failure. *European Journal of Information Systems* 5:34–46.

Puri, S. K., Chauhan, K. P. S., and Admedullah, M. 2000. Prospects of biological diversity information management. In *Information flows, local improvisations and work practices*. Proceedings of the IFIP WG9.4 Conference 2000. Cape Town: IFIP.

Robey, D., and Boudreau, M.-C. 1999. Accounting for the contradictory organizational consequences of information technology: Theoretical directions and methodological implications. *Information Systems Research* 10(2):167–185.

Roche, E., and Blaine, M., eds. 1996. *Information technology, development and policy*. Aldershot, UK: Avebury.

Sauer, C. 1993. *Why information systems fail: A case study approach*. Henley, UK: Alfred Waller.

Sauer, C. 1999. Deciding the future for IS failures: Not the choice you might think. In *Rethinking management information systems*, eds. R. Galliers and W. L. Currie, pp. 279–309. Oxford, UK: Oxford University Press.

Scott Morton, M., ed. 1991. *The corporation of the 1990s: Information technology and organizational transformation*. Oxford, UK: Oxford University Press.

Shields, P., and Servaes, J. 1989. The impact of the transfer of information technology on development. *The Information Society* 6(1–2):47–57.

Shukla, M. 1997. *Competing through knowledge*. New Delhi: Response Books.

Silva, L. O., Castro, J. C., and Rodriguez, E. O. 2000. Outsourcing as an improvisation. In *Information flows, local improvisations and work practices*. Proceedings of the IFIP WG9.4 Conference 2000. Cape Town: IFIP.

Southon, G., Sauer, C., and Dampney, K. 1997. *Lessons from a failed information systems initiative: Issues for complex organisations*. Paper presented at APAMI/HIC'97 Conference, Sydney, Australia.

Suchman, L. 1987. *Plans and situated actions*. Cambridge, UK: Cambridge University Press.

Tettey, W. J. 2000. Computerization, institutional maturation and qualitative change. *Information Technology for Development* 9(2):59–76.

Venkatraman, N. 1989. The concept of fit in strategy research. *Academy of Management Review* 14(3):423–444.

Walsham, G. 2000. IT, globalisation and cultural diversity. In *Information technology in context*, eds. C. Avgerou and G. Walsham, pp. 291–303. Aldershot, UK: Ashgate.

Wilson, G., and Heeks, R. 2000. Technology, poverty and development. In *Poverty and development into the 21st century*, eds. T. Allen and A. Thomas, pp. 403–424. Oxford, UK: Oxford University Press.

World Bank. 1993. *Turkey: Informatics and economic modernization*. Washington, DC: World Bank.

Wynn, E. and deLyra, J. 2000. A strange attractor in the chaos of global IS development. In *Information flows, local improvisations and work practices*. Proceedings of the IFIP WG9.4 Conference 2000. Cape Town: IFIP.