

11 SUPPORTING ENGINEERING OF INFORMATION SYSTEMS IN EMERGENT ORGANIZATIONS

Sandeep Purao

The Pennsylvania State University

Duane P. Truex III

Florida International University and

Georgia State University

Abstract

Research related to information systems development has roughly followed two diverse paths. The first, pursued by the software engineering community, is aimed at creating techniques for the efficient engineering of IT artifacts. The second, pursued by communities such as IFIP 8.2, attempts to understand and anticipate the impact of IT on organizations or upon one another by incorporating social science theories. The two views are in conflict because the former treats information systems merely as reflections of requirements, whereas the latter views them as agents of change. As a result, the two streams have suffered from increasing emphasis on minutiae and are, at worst, in danger of losing their relevance. Recent research in emergent systems development and developers' engagement in problem and design spaces suggest a possible approach to integrating the two streams. In particular, we argue that novel R-forms (representation techniques) can proactively facilitate the engineering of information systems in emergent organizations. Using insights from research in both streams, we develop a set of requirements that can guide the development of new R-forms that may take into account both the engineering of the IT artifact as well as the emergent nature of organizational context in which the IT artifact will be deployed.

Keywords: Emergent systems design, deferred systems design, social theories, information systems development (ISD), representational forms

1 INTRODUCTION

Much current research in software engineering focuses on models and methods for creating IT artifacts that are intended to function in organizational contexts. An implicit

assumption in this research stream is the availability of stable or slowly evolving requirements (Robinson et al. 2003) and the need to reflect these accurately in the object system being constructed (Welke 1980). Improved methods and models following this assumption cannot sufficiently address demands posed by increasingly turbulent business environments, which require not only changes to the IT artifacts but also business practices. Such traditional ideals of information systems development are being questioned by researchers in the IFIP WG 8.2 community (e.g., emergent systems development—Bello et al. 2002; Truex, Baskerville, and Klein 1999; amethodical systems development—Truex, Baskerville, and Travis 2000; deferred systems development—Patel 1999; and improvisation and bricolage—Bansler and Havn 2002). The IFIP Working Group 8.2 focus may be described as making sense of the complex interrelationship between the IT artifact and the organizational context in which it is embedded. Much of this work has been conducted as *post hoc* studies of impacts following the implementation of information systems. Resulting insights from this community, while often telling and profound, have not been translated into proactive, actionable techniques for building more effective information systems. Thus, the two related research communities, each ostensibly having the mission to help design and introduce effective information systems into organizational settings, have missed important opportunities to interact and work on this shared goal. Without greater awareness and integration of the work in the two domains, there is an increasing risk that each will continue its emphasis on minutia accompanied by decreasing relevance of the research outcomes. This paper builds its argument in the context of repeated calls for integrating insights from these two research streams (see, for example, Probert and Rogers 1999).

A specific objective of the paper is to investigate a possible path to realizing this integration in the form of representation techniques that can assist in codesigning information systems and emergent organizational forms. The remainder of the paper is organized in four sections. Section 2 traces research in the two streams identified above with a view to highlighting the need for integration. In section 3, we envision how modeling techniques may be used to facilitate this integrated perspective (i.e., for codesigning information systems and emergent organizations following developer behaviors observed in practice). Section 4 develops desired requirements for such representational techniques. We conclude in section 5.

2. PRIOR RESEARCH

2.1 A Software Engineering Perspective

The software engineering perspective views information systems as technical artifacts that need to be developed, built, and deployed. It, therefore, requires a focus on techniques and models that facilitate design and production of software artifacts with a limited consideration of the environments in which the artifact will be deployed. Pressman and Associates (2003) define software engineering as a discipline that encompasses the process associated with software development, the methods used to analyze, design, and test computer software, the management techniques associated with

the control and monitoring of software projects, and the tools used to support process, methods, and techniques. Like many other engineering disciplines, the environment (e.g., an organization) is seen as the source of requirements, often simplified to organizational processes, behaviors, or roles (ignoring important issues such as power, conflict, or structure). The goal of software engineering is, thus, converting these requirements to a format appropriate for realization in the software artifact in a manner that leads to few bugs, on time delivery, and greater maintainability. More enlightened approaches from the requirements engineering research community (see, for example, Nuseibeh 2001; Nuseibeh and Easterbrook 2000) have argued for a closer connection between real-world *goals* and *software specifications* favoring a multidisciplinary, human-centered process for requirements engineering (Nuseibeh and Easterbrook 2000, p. 38). Such linkage has also been explored for the purpose of understanding nonfunctional requirements (Mylopoulos et al. 1992) and for expressing the mappings between different viewpoints during the specification and capture of requirements (Nuseibeh et al. 1994).

Methods and modeling techniques in software engineering, however, continue to assume relative stability and consensus in an organization. The techniques often simplify, generalize, and abstract the organizational domain to develop archetypes, which are then translated into functional descriptions (Gane and Sarson 1979) or use cases (OMG 2003). These plan-driven methodologies (Boehm 2002) are often not appropriate for turbulent environments, where business practices or technology is undergoing rapid change. Levine et al. (2002), for example, suggest that requirements for many systems are vague and likely to change. Truex, Baskerville and Klein (1999) suggest that developing a set of specifications that are clear, consistent, and complete before starting the design may be a fools errand. As a response, the definition of requirements specifications has been expanded to include evolution over time and across software families (Zave 1997). However, the linkage between the organizational subsystem and the information system is still tenuous, largely restricted to treating the former as the source of requirements.

One large-scale effort addressing the formalization of concepts for organizational information systems has been the FRISCO project (1997, 2001), representing the culmination of a decade-long multinational research effort. The FRISCO report develops an elaborate system of definitions to map underlying conceptions of things in a domain to representations of those concepts (FRISCO 1997, p. 34). The superstructure of definitions in FRISCO has been critiqued by Hesse and Verrijn-Stuart (2000) for reasons such as over-formalization, circular definitions, and ambiguity. Similarly, Stamper (2000), one of the signatories to the FRISCO report, has argued forcefully for the combination of signs and norms as the basis for a better conceptualization of organizational information systems in his critique of the FRISCO project and in reports on the MEASUR project (1994). More recently, the notion of complete requirements specifications drawn from largely static organizational perspectives is being challenged by lightweight or agile methodologies (Cockburn 2001), extreme programming (Beck 2000), SCRUM (Schwaber and Beedle 2002), and Crystal (Cockburn 2001). Another technique proposed to manage software evolution in changing environments is traceability (i.e., linking requirements to the artifacts being created; Ramesh and Jarke 2001). However, even these approaches underplay the organizational context, providing

no explicit mechanisms to take into account concerns such as conflict or structural changes in the organization that may be brought about by the introduction of the information system.

As this overview suggests much work in software engineering is motivated by the idea that changes introduced late in the software development cycle are costly and should be avoided as much as possible. Information systems, thus, are seen as reflections of requirements instead of as agents of change for the intended environment. As reflections of complete requirements, the systems are expected to be relatively stable, resembling the computational view of the IT artifact described by Orlikowski and Iacono (2001).

2.2 A Social Science Perspective

A social science perspective, on the other hand, considers information systems as agents of change and not mere technical artifacts. Practitioners sharing this worldview consider the insertion of information systems in organizations as opportunities to study organizational consequences of the introduction of the artifact—following a nominal view of the IT artifact (Orlikowski and Iacono 2001). A number of these researchers using structuration theory also investigate the dualistic interplay of the organization and technology (DeSanctis and Poole 1994; Jones 1999; Lyytinen and Ngwenyama 1992; Orlikowski 1992; Orlikowski and Robey 1991; Truex, Baskerville, and Travis 2000; Walsham 2002; Walsham and Han 1991). Others using critical social theoretic approaches examine the roles of power differential and how IT emancipates or imprisons social actors. Still others, guided by actor network approaches examine the development and interactions of human and nonhuman networks. The thrust of this perspective is, therefore, on understanding and explaining the underlying logic of social organizations when they come into contact with information systems.

Jones (2000), based on a content analysis on the IFIP 8.2 working group proceedings, identifies predominant social science theories as Giddens' (1979, 1984, 1987) structuration theory, Habermas' (1984) theory of communicative action, and the actor-network theory (Latour 1987, 1993, 1996, 1998)—which attests to the complex relationship between the organizational context and the IT artifact. Further bolstered in terms of their appropriateness by Holmström and Truex (2001), these represent appropriate theoretical lenses to study organizational changes. Giddens' structuration theory, for example, is considered well-suited for exploring the ephemeral boundaries between IT and organization, characterizing IT as both enabler and constrainer of social action (DeSanctis and Poole 1994; Orlikowski 1992, 1996, 2000; Orlikowski and Robey 1991). Other social theories address related questions, but from different points of view. Critical social theory as applied to IS research addresses issues of conflict and of power differentials in ISD and IS use (Apel 1980; Bourdieu 1991; Habermas 1981, 1984). Actor-network theory addresses the complex relationships between human and non-human actants in rich social and technological and social domains (Callon 1986, 1991; Latour 1987, 1993, 1996, 1998). The challenge for these research streams is to go beyond the idea of reactively studying information systems as change agents to proactively improving specific ways of engineering systems that can contribute to the desired changes in the environment.

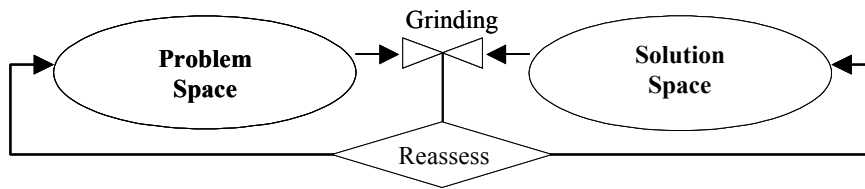


Figure 1. Heterogeneous Engineering (Adapted from Bergman et al. 2001)

2.3 Reconciling the Perspectives

Both the software engineering and organizational IS communities have been slow at recognizing that the unidimensional views embraced within their own research stream may be inadequate to deal with the difficult problems associated with developing and deploying complex information systems that must function within emergent organizations. Even recent articles defining the boundaries of IS research and of the conditions that define IS research often narrowly define the field, leaving aspects of design and human research out of the description (Benbasat and Zmud 2003; Weber 2003a, 2003b). A few recent writings, however, attest to the increasing recognition of the limits of these unidimensional approaches. Bergman et al. (2001), for example, suggest the term *heterogeneous engineering*, that is, requirements analysis as an iterative process moving between an existing solution space to problem space, then back again to the future solution space, a process they metaphorically call *grinding* (see Figure 1).

They recognize at a macro scale the interplay between the information system and the organization captured at the micro level (Guindon 1990; Mathiassen and Munk-Madsen 2000; Purao, Rossi, and Bush 2002) with the ideas of *problem* and *design* spaces. We note that this view presents an epistemological challenge in that it separates the social context (the organization) and the technical component (the IT artifact). An alternative is to see the two as parts of a whole, wherein the IT artifact becomes a part of the organizational fabric and where organizational assumptions are woven into the IT artifact. Viewing the two objects separately suggests a dualism versus seeing the duality of the organization and the IT artifact. A possible approach to this quandary is seen in the line of work loosely called emergent systems development or deferred systems development that is challenging traditional notions of systems development (Bansler et al. 2000; Bansler and Havn 2002; Bello et al. 2002; Patel 1999; Truex, Baskerville, and Klein 1999; Truex, Baskerville, and Travis 2000). This work suggests a fundamental shift from traditional development practices to a continuous redevelopment process (Truex, Baskerville, and Klein 1999).

The need to reconcile the two perspectives is also seen in the work of Liu (2000), and Liu et al. (2002a), which suggests use of semiotics for information systems engineering. Liu et al. (2002b) also argue for the need for codesigning information systems and organizational processes, presenting the hypothesis that “an organic integration of IT into both processes will allow both systems to evolve naturally” (p. 254), emphasizing that the codesigning of business and IT systems is an important research issue. De Moor (2002) suggests a requirements elicitation process that may

allow surfacing of concerns and constraints to inform such codesign. Eatock et al. (2002) demonstrate the possibility of using simulation to study the problem, whereas Beeson et al. (2002) describe a case study focused on identifying and modeling links between information systems and business. These efforts are also indicative of the early research efforts in this emerging research area.

As a group, these authors further underscore the need to reconcile two research streams. Although the notions of “IT artifact as a reflection of requirements” and “IT artifact as change agent” may, *prima facie*, seem incommensurate, the above writings point to several possibilities for reconciling the two perspectives. Specifically, the line of work on deferred system development challenges the software engineering perspective away from its focus on the notion of reflecting requirements while recognizing the emergent nature of organizations. A specific approach to addressing this is to support the engineering of software in emergent organizations in the form of representations that may encompass both perspectives.

3. SUPPORTING THE ENGINEERING OF INFORMATION SYSTEMS IN EMERGENT ORGANIZATIONS

The engineering of software for emergent organizations presents several interlinked challenges. On one hand, the very idea of engineering an IT artifact requires us to confer upon it an ontological status that is separate from the organization in which it will be deployed. On the other hand, considering the organization as emergent requires that we consider the mutually interdependent nature of the IT artifact and the organization, removing the distinct ontological status accorded to it for the purpose of engineering. This mirrors the epistemological challenge described earlier, that is, *engineering in emergent organizations* requires us to simultaneously separate the social context (the organization) and the technical component (the IT artifact), as well as consider it as a duality, where the IT artifact is a part of the organizational fabric with organizational assumptions woven into the IT artifact.

Thus, we struggle with how we can facilitate such an integrated, yet discrete, perspective on IT artifacts and organizations. We believe that a prerequisite to reconciling these viewpoints is to consider each—the IT artifact as well as the organization—as malleable and being in a constant state of flux. The process of emergence (Truex, Baskerville, and Klein 1999), then, must be seen as one that is aimed at codesigning both the information systems and the organization. We note that codesigning refers to a process, not a specific outcome, that is, the process of codesigning may result in a series of temporary equilibria, which hold in balance the IT artifact and the organization. Facilitating a process that allows such codesigning also requires that we take into account developer behaviors observed in practice (Purao, Rossi, and Bush 2002), which point to cycles between the problem and design space (i.e., the organization and the IT artifact). The cycles are punctuated by apparent resting points or emergent regularities that allow the organization to function normally, that is, at times of relative regularity when the distance between the design and problem spaces may be comfortably slight.

3.1 The Role of Representations

As in other professional disciplines, both ISD research communities embrace the need to create representations of extant and changed *object* systems. Unlike architecture or engineering, however, representations play an even more vital role for information systems development for two reasons. The artifacts developed by information systems developers are virtual as opposed to real, making representations extremely important as a mode of communication. And the environments where these artifacts are intended to be deployed are themselves malleable, making the representations of interaction between the two even more important.

A modeling technique, in essence, externalizes representations that cannot be “held in the head” by a developer, allowing the developer to extend the short-term memory to notations that spill over into the physical world. The world being modeled, in all but the most trivial examples, is simply too complex; often it is too complex to be modeled by a single person. This requires then the sharing and codevelopment of the representation, requiring close interaction and communication between those creating the representation. The representations must, therefore, support effective communication between different participants in the development process. A representational technique can be used, for example, to envision a system, and its behavior, before specifying the system (for example, scenario-based design; Carroll 1995). Representation techniques are also important because of the significant role they play in problem-solving and development as aids to cognition (Bødker 1998; Suchman 1990). Better representation has been shown to directly contribute to better solutions by bringing to light solution approaches heretofore not evident (Larkin and Simon 1987). Finally, representation techniques are also important because they suggest to the developer what is most important and relevant, and what may be ignored. The absence of a technique for codesigning the organization and the artifact, therefore, implicitly suggests that organizational context is not important. Clearly this is an untenable position.

3.2 Representations to Support Engineering of Software

A large number of representation techniques have been proposed for the engineering of software. A representative example is found in the unified modeling language (UML), which suggests modeling the structural, behavioral, dynamic, and functional properties of IT artifacts. A survey by Wieringa (1998) classifies and discusses these techniques in detail. Several other techniques have been proposed over the years including the so-called traditional techniques such as the data flow diagram (Gane and Sarson 1979), entity-relationship modeling (Chen 1976) and numerous extensions thereof, as well as newer modeling techniques for object-oriented modeling such as UML and OML (Opdahl et al. 2000). Reviews and classifications of these techniques appear in Rossi and Brinkkemper (1996) and Siau et al. (1997). In addition, considerable work has been carried out in understanding appropriateness of modeling notations (Kim et al. 2000), modeling the modeling notations (Rossi et al. 1992), and evaluation of modeling notations (Wand and Weber 1993).

In their research manifesto for conceptual modeling Wand and Weber (2002) suggest four levels of focus for research activities. These include grammar, methods,

scripts, and context. They indicate that while work has been carried out in the first two levels, much less work has been carried out for the last two. Work in the first stream has dealt with issues such as faithful representation, ontological expressiveness, and overlaps. Work in the second stream has dealt with issues of appropriate application of the grammar. Work in the third and fourth streams has dealt with the idea of using the grammars to generate scripts, and the use of conceptual modeling techniques in different contexts, different individuals, and their empirical investigation. The agenda presented by Wand and Weber (2002) has, at the core, the goal of accurate reflection of the universe of discourse (see section 2.1). They indicate, for example, their key motivation as “how can we model the world to better facilitate...information systems?” (Wand and Weber 2002, p. 363). The arguments we have presented so far require us to go beyond this perspective, treating the organization itself as emergent.

3.3 Representations to Describe Emergent Organizations

Few representation techniques have been proposed for modeling organizations in the context of designing information systems. Where such techniques have been proposed, they have been infused with metaphors borrowed from software engineering, imposed on the organizational practices and processes. For example, as early as 1979, data-flow-diagrams were proposed to model current business functions (Gane and Sarson 1979). Techniques such as BIAIT and PSL/PSA attempted to match the complex relationships between data, process, and owners of both in organizations. More recently, the IDEF suite of process maps (Mayer et al. 1995) has been proposed. These techniques have not directly addressed purely organizational concepts, instead focusing on only those aspects that will eventually be represented in an information system.

In contrast to these software engineering techniques are graphical approaches like rich pictures as devised by Checkland (1998, 1999) and adopted in Multiview by Avison and Wood-Harper (1990), and others in which cartoon-like drawings provide high-level representations identifying stakeholders, their concerns, and some of the structure underlying the work context. This technique remains informal, providing no direct mapping from the rich picture to other representation forms and models of data, workflow, human interaction, or organizational process. Returning to textual representations, Boland (1985, 1991) and others have recognized the importance of the concept *metaphor* when deployed as a grammatical construct used in describing organizational context and ambiguity.

However, bridging from rich organizational contexts to the information system being developed has proved an elusive goal for several reasons. Organizations themselves are ephemeral and emergent. Processes and the interaction of human and machine actors are never really fixed. They are, in short, prototypical dynamic organisms, devilishly difficult to model and replicate. Scenarios are limited because each provides a representation of an organizational moment frozen in time. This technique offers no direct way to stitch the scenarios together to represent a continuously changing organizational tapestry. While scenarios offer the advantage of “ready to go” models, like all contingency approaches, anticipating when to deploy one scenario versus another and knowing in what particular order remains an inexact science.

3.4 Representations that Support Developer Engagement with Both Domains

The usefulness of a representation technique is moderated by how well it supports behaviors that developers may utilize in their development processes. A few prescriptive descriptions (e.g., Kruchten 1998) have attempted to specify correct modes of using representation techniques. However, a more practice-oriented view is necessary to fully understand how representation techniques may help developer behaviors. Further, it is necessary to understand how developers engage with both the envisioned artifact as well as the environment where the artifact is intended to be deployed even when there are no formal representation techniques at hand. There is some evidence that a systems analyst's mental models, experience, and training help address this challenge (Fitzgerald 1997; Lee and Truex 2000) but this dependence on non-systematic factors does not lead to the kind of replicability sought by creators of formal methods. Having representations to help in the process is therefore thought to advance the discipline in a more orderly fashion.

A recent study by Purao, Rossi, and Bush (2002) identifies specific behaviors that developers use for engaging with problem and design spaces, which loosely mirror the ideas of the organization, and the IT artifact. Their study reveals that developers do, indeed, treat both spaces as malleable and engage with each either in turn or simultaneously, and represent elements of each that affect their decisions in the other. They found evidence in developer behaviors that may be interpreted as indicating strong support for the duality between the information system and the domain of influence, which they term the problem space and the design space. Micro-cycles between the two spaces appear as back-and-forth engagements in the two spaces, which facilitate simulation and expansion of the design space. Problem space behaviors represent reinterpreting the problem space based on decisions in the design space. These three directions—engineering of software, describing the emergent organization, and accommodating developer behaviors—suggest the need for novel representational forms (or R-forms; Welke 1980) that can support engineering the emergence of both, the IT artifact as well as the organization in which it will be deployed.

4. ENGINEERING THE EMERGENCE: AN R-FORMS MANIFESTO

We argue that new R-forms (modeling and representation techniques; Welke 1980) are necessary to support the codesigning of information systems and organization (i.e., to support the three directions described above). Clearly, such a technique must not sacrifice the first direction, engineering of software, which has been traditionally the primary consideration for representational forms. Rather, it should strive to incorporate constructs or means to support the second dimension, portraying organizational emergence. Finally, it must facilitate and support the third direction, known developer behaviors such as cycles of engagement with the two spaces. Desirable attributes of such R-forms are difficult to identify because they require operationalizing micro-level

behaviors and contextually mandated adjustments of the type outlined above. Such operationalizing will, of necessity, involve some over-determination reflecting the researchers' biases. Therefore, instead of presenting prescriptions that will dictate specific structures or procedures that may shape such R-forms, we outline a set of requirements that could guide research interested in the discovery and assessment of such R-forms. Our recommendations are guided by the following key shortcomings of existing R-forms.

- An implicit, exclusive focus on facilitating the engineering of the IT artifact
- An implicit mode of representing frozen moments in time, favoring a snapshot view instead of an evolving perspective
- A codification of universal descriptions of work practices and processes instead of recognizing the particulars and context-laden specializations
- A fixation on issues such as *correctness* and *completeness* favoring the assumption of a closed set of the functionality of the IT artifact
- An implicit operationalization of the duality between the organization and the IT artifact instead of recognizing the dualism that the two represent
- Treating only the IT artifact as malleable during the development process and the organization as rigid

It is necessary to examine whether we have a paradox in the making specifically with our call for R-forms to address the third shortcoming indicated above. For example, it has been argued (Truex, Baskerville, and Klein 1999; Truex, Baskerville, and Travis 2000) that standard, universal, one-size-fits-all solutions are not appropriate for organizational life, which is complex and can often present a series of unique challenges. Current prescriptions for R-forms, on the other hand, require classification, generalization, and specification of constructs that may be embedded in the IT artifact. Is it possible, then, to reconcile the concept of uniqueness and organizational emergence against that of classification, embedded in the current R-forms, which allow IT artifacts to be engineered? A fully developed and cogent answer will, in fact, take the shape of a novel R-form. One possible approach to reconciling the two is suggested by Parsons and Wand (2000) in the context of database design, where they separate the specification of the schema layer and the instance layer, allowing varying mapping between the two. Work related to schema evolution in object-oriented databases (Franconi et al. 2000) also suggests that it is, in fact, possible to accord instances a first-class citizenship. These efforts suggest that it is possible to construct IT artifacts that respect the particularities of organizational life.

In particular, realizing such an R-form will require that we take account of a number of organizational concepts familiar to the IFIP WG 8.2 community. Those concepts include (1) the relative persistence of certain social structures when reinforced by technologies supporting dominant values and power relationships, (2) the interplay and relative agency of both human and organizational actors, (3) how systems may be used to consolidate power and control or to emancipate, and (4) the process of negotiation enabling constant changes in organizational process and meanings. Doubtless there are other such notions requiring attention. IS researchers have identified theories and have advanced the discourse on these topics. We have found ways to examine and describe

these concepts in the interplay of ISD and IS use in organizational life, but not to model them effectively or parsimoniously. It is difficult to capture these concepts because they represent ephemeral and dynamic social processes. Even so, they should be represented in our descriptions of systems.

4.1 R-Forms: A Set of Requirements

Based on the discussion above, we outline a set of requirements for R-forms. Where possible, we indicate ongoing work that may be leveraged to help creation of such R-forms.

- **Separating universals and particulars.** The R-form should allow modeling of categories and classes as well as instances that may not fit into the categories. A similar case is made by Parsons and Wand (2000) in their suggestions for a new mode of database design that separates schemas and instances.
- **Portraying multiple perspectives.** The R-form should allow multiple perspectives to deal with the underlying complexity of the phenomenon—the IT artifact as well as the organizational environment in which it will function. A similar use of multiple perspectives is seen in the now *de facto* standard for modeling object-oriented software, UML (OMG 2003).
- **Capturing organizational characteristics.** The R-form should allow capturing organizational actors, relationships among them, and fuzzy notions such as power. Techniques analogous to these are found in the portrayal of roles in workflow diagrams (van der Aalst et al. 2003), and those for illustrating webs of power and influence and interaction (Kling 1987; Kling and Scacchi 1982). Early work related to rich pictures represents another direction that can be useful here.
- **Balancing formal and informal descriptions.** The R-form should not only possess sufficient expressive power to capture, informally, the organizational characteristics, but should also contain sufficient formalisms to capture the details needed for engineering the software solutions. The balance between formal and informal descriptions is also necessary to ensure ease of use for the developers.
- **Bridging the duality between the IT artifact and the organization.** The R-form should allow mechanisms to ensure that elements from the IT artifact can be mapped against those in the organization and vice versa to ensure internal consistency. A similar requirement for consistency is enforced by UML modeling conventions (OMG 2003). Such conceptual bridging should ensure that both, the IT artifact itself, as well as the organization, is treated as malleable, allowing codesign of both. One technique addressing this goal is suggested in recent work by Purao, Truex, and Cao (2003).
- **Allowing design to continue beyond deployment.** The R-form should recognize that emergence must continue past deployment of the information system, and that the process of designing should continue to encompass what is traditionally referred to as the maintenance stage. A possible approach to achieving this suggested by Chua et al. (2003).
- **Supporting progress without commitment.** The R-form should allow capturing incomplete decisions allowing progress without commitment. A similar idea is put

forward in the context of use cases (see survey by Hurlbut 2003), which allow specifications to evolve without requiring commitment.

- **Managing complexity without defining it away.** It should allow means to simultaneously represent the whole of complex settings while allowing users to zero in on elements requiring immediate focus. This is analogous to the decomposition allowed in traditional data-flow diagramming. On the one hand, the analyst should be able to represent the whole of the system, perhaps in a rich picture form and follow with a logical mapping through successive layers of detail to the desired level and objects—either technical or organizational—of interest.

Interesting conceptual advances have been made in a number of fields that may lead to possible solutions that meet the above requirements. We have already pointed out several advances in the discussion above. A complete explication of these is beyond the purview of this paper, and continues to be part of our ongoing research agenda. A few promising directions we are considering include the ideas of patterns (Fowler 1997), metaphors (Lakoff and Johnson 2003; Madsen 1994), and scenarios (Carroll 1995) and in the work of those exploring deferred systems development approaches.

5. CONCLUDING REMARKS

Combining insights from software engineering and social theories, we have argued that new R-forms are necessary to support the codesign and coevolution of problem and design spaces. In doing so, we are presenting a counterargument to that presented by Wand and Weber (2002). Their agenda focuses solely on the concern of accurate reflection of requirements, which they indicate as “how can we model the world better to facilitate the development of information systems?” The agenda we have argued for in this paper suggests expanding the set of concerns to address the question, how can we model the envisioned IT artifact as well as the context in which it will be deployed better to support the codesign and coevolution of both, the IT artifact as well as the organizational context?

We have positioned our arguments in a manner that does not detract from the notion of engineering IT artifacts, which we believe continues to be an important dimension of the systems development process. This dimension, however, does not take account of the symbiotic relationship between the IT artifact and the organizational context—a relationship that is better described as emergence. In our arguments, we have attempted to integrate these orthogonal dimensions to argue for novel R-forms that may better support engineering of IT artifacts in emergent organizations.

As of this writing there are no fora dedicated to the goal of reconciling these important and complex questions. Thus we turn to the IFIP WG 8.2 community, which in its earliest days chose to cast the gauntlet to the establishment with a call for the IS research community and for the establishment within that community to consider more expansive and less familiar ways by which one could consider and research IS phenomenon and open minds as to the kinds of ontological and epistemological questions that would be relevant to such studies. The IFIP WG 8.2 community has demonstrated its ability to embrace new and challenging approaches to IS research. We

hope they are willing and able to take the challenge we have outlined and offer both advice as to how to advance the discourse and a forum in which it may be developed.

REFERENCES

- Apel, K.-O. *Towards A Transformation of Philosophy*, Translated by G. Adey and D. Frisby, Edited by John O'Neill, London: Routledge & Kegan Paul, 1980.
- Avison, D. E., and Wood-Harper, A. T. *MULTIVIEW: An Exploration in Information Systems Development*, Oxford: Blackwell Scientific Publications, 1990.
- Bansler, J. P.; Damsgaard, J.; Scheepers, R.; Havn, E.; and Thommesen, J. "Corporate Intranet Implementation: Managing Emergent Technologies and Organizational Practices," *Journal of the Association of Information Systems* (1), December 2000.
- Bansler, J. P., and Havn, E. "Improvisation and Bricolage in Information Systems Development: A Field Study," Working Paper 11-11-02, 10, Technical University of Denmark, Copenhagen, 2002.
- Beck, K. *eXtreme Programming Explained*, Reading, MA: Addison Wesley, 2000.
- Beeson, I.; Green, S.; Sa, J.; and Sully, A. "Linking Business Processes and Information Systems Provision in a Dynamic Environment," *Information Systems Frontiers* (4:3), 2002, pp. 317-329.
- Bello, M.; Sorrentino, M.; and Virili, F. "Web Services and Emergent Organizations: Opportunities and Challenges for IS Development," in S. Wrycza (Ed.), *Proceedings of the 10th European Conference on Information Systems*, Gdansk, Poland, 2002, pp. 439-449.
- Benbasat, I., and Zmud, R. W. "The Identity Crisis within the IS Discipline: Defining and Communicating the Discipline's Core Properties," *MIS Quarterly* (27:2), June 2003, pp. 183-194.
- Bergman, M.; King, J.; and Lyytinen, K. "Large Scale Requirements Analysis as Heterogeneous Engineering," in C. Floyd and R. Klischewski (Eds.), *Social Thinking—Software Practice*, Cambridge, MA: MIT Press, 2001.
- Bødker, S. "Understanding Representation in Design," *Human-Computer Interaction* (13:2), 1998, pp. 107-125.
- Boehm, B. "Agile and Plan-Driven Methodologies: Oil and Water," Presentation at Agile Universe 2002, August 5, 2002 (available online at <http://www.agilealliance.com/articles/articles/agileAndPlanDrivenMethods.pdf>; accessed December 30, 2003).
- Boland, Jr., R. J. "Phenomenology: A Preferred Approach to Research on Information Systems," in E. Mumford, R. Hirschheim, G. Fitzgerald, and A. T. Wood-Harper (Eds.), *Research Systems in Information Systems*, Amsterdam: North-Holland, 1985, pp. 193-201.
- Boland, R. "Information Systems Use as a Hermeneutic Process," in H.-E., Nissen, H. K. Klein, and R. Hirschheim (Eds.), *Information Systems Research: Contemporary Approaches & Emergent Traditions*, Amsterdam: North-Holland, 1991, pp. 439-458.
- Bourdieu, P. *Language and Symbolic Power*, Cambridge, MA: Harvard University Press, 1991.
- Callon, M. "Some Elements of a Sociology of Translation: Domestication of the Scallops and the Fishermen of St Briec Bay," in J. Law (Ed.), *Power Action and Belief: A New Sociology of Knowledge*, London: Routledge, 1986, 196-233.
- Callon, M. "Techno-Economic Networks and Irreversibility," in J. Law (Ed.), *A Sociology of Monsters: Essays on Power, Technology and Domination*, London: Routledge, 1991, 132-161.
- Carroll, J. M. *Scenario Based Design*, New York: Wiley & Sons, 1995.
- Checkland, P. *Systems Thinking, Systems Practice*, New York: John Wiley & Sons, Ltd., 1999.
- Checkland, P., and Holwell, S. *Information, Systems, and Information Systems: Making Sense of the Field*, Chichester, England: John Wiley & Sons, 1998.

- Chen, P. P. S. "The Entity-Relationship Model: Towards a Unified View of Data," *ACM Transactions on Database Systems* (1:1), 1976, pp. 9-36.
- Chua, C. E. H.; Puroo, S.; and Storey, V. "An Approach for Developing Maintainable Software," in J. Ross and D. Galletta (Eds.), *Proceedings of the 9th Americas Conference on Information Systems*, Tampa Florida, 2003, pp. 1911-1921.
- Cockburn, A. "Agile Software Development Ecosystems," in A. Cockburn and J. Highsmith (Eds.), *The Agile Software Development Series*, Reading, MA: Addison Wesley Longman, 2001.
- De Moor, A. "Language/Action Meets Organizational Semiotics: Situating Conversations with Norms," *Information Systems Frontiers* (4:3), 2002, pp. 257-272.
- DeSanctis, G., and Poole, M. S. "Capturing the Complexity in Advanced Technology Use: Adaptive Structuration Theory," *Organization Science* (5:2), 1994, pp. 121-147.
- Eatock, J.; Paul, R. J.; and Serrano, A. "Developing a Theory to Explain the Insights Gained Concerning Information Systems and Business Process Behavior: The ASSESS-IT Project," *Information Systems Frontiers* (4:3), 2002, pp. 303-316.
- Fitzgerald, B. "The Use of Systems Development Methodologies in Practice: A Field Study," *Information Systems Journal* (7), 1997, pp. 201-212.
- Fowler, M. *Analysis Patterns: Reusable Object Models*, Reading, MA: Addison Wesley, 1997.
- Franconi, E.; Grandi, F.; and Mandreoli, F. "A Semantic Approach for Schema Evolution and Versioning in Object-Oriented Databases," *Computational Logic 2000*, London, July 24-28, 2000.
- FRISCO. "A Framework of Information Systems Concepts: The FRISCO Report," 1997 (available online at <http://www.wi.leidenuniv.nl/~verrynst/fri-full-7.pdf>).
- FRISCO "A Framework of Information System Concepts: The Revised Frisco Report," Draft January 2001 (available online at <http://www.wi.leidenuniv.nl/~verrynst/Draft-Jan-01.zip>).
- Gane, E. C., and Sarson, T. *Structured Systems Analysis: Tools and Techniques*, Englewood Cliffs, NJ: Prentice-Hall, 1979.
- Giddens, A. *A Contemporary Critique of Historical Materialism*, Volume 2, Berkeley, CA: University of California Press, 1987.
- Giddens, A. *Central Problems in Social Theory: Action, Structure and Contradiction in Social Analysis*, Berkeley, CA: University of California Press, 1979.
- Giddens, A. *The Constitution of Society: Outline of the Theory of Structuration*, Cambridge, MA: Polity Press, 1984.
- Guindon, R. "Designing the Design Process: Exploiting Opportunistic Thoughts," *Human-Computer Interaction* (5:2-3), 1990, pp. 305-344.
- Habermas, J. "Reason and the Rationalization of Society," Chapter 3 in *The Theory of Communicative Action*, Boston: Beacon Press, 1981, pp. 273-333.
- Habermas, J. *The Theory of Communicative Action*, Volume 1, Boston: Beacon Press, 1984.
- Hesse, W.; and Verrijn-Stuart, A. A. "Towards a Theory of Information Systems: The FRISCO Approach," in *Proceedings of the 10th European-Japanese Conference on Information Modeling and Knowledge Bases*, Saariselkä, Finland, 2000.
- Holmström, J., and Truex, D. "What Does it Mean to be an Informed IS Researcher? Some Criteria for the Selection and Use of Social Theories in IS Research," in S. Bjørnstad, R. E. Moe, A. I. Mørch, and A. L. Opdahl (Eds.), *Proceedings of the Information Systems Research in Scandinavia 2001 Conference*, Volume 2, Bergen, Norway, August 11-14, 2001, pp. 313-326.
- Hurlbut, R. *A Survey of Approaches For Describing and Formalizing Use Cases*, Wheaton, IL: Expertech, Ltd., 2003 (available online at <http://www.iit.edu/~rhurlbut/xpt-tr-97-03.html>; accessed August 20, 2003).
- Jones, M. "The Moving Finger: The Use of Social Theory in WG 8.2 Conference Papers, 1975-1999," in R. Baskerville, J. Stage, and J. I. DeGross (Eds.), *Organizational and Social*

- Perspectives on Information Technology*, Boston: Kluwer Academic Publishers, 2000, pp. 15-31.
- Jones, M. "Structuration Theory," in W. L. Currie and R. D. Galliers (Eds.), *Re-Thinking Management Information Systems*, Oxford: Oxford University Press, 1999, pp. 103-135.
- Kim, J.; Hahn, J.; and Hahn, H. "How Do We Understand a System with (So) Many Diagrams? Cognitive Integration Processes in Diagrammatic Reasoning," *Information Systems Research* (11:3), 11:3, September 2000, pp. 284-303.
- Kling, R. "Computerization as an Ongoing Social and Political Process," in G. Bjerknes, P. Ehn, and M. Kyng (Eds.), *Computers and Democracy: A Scandinavian Challenge*, Aldershot, England: Avebury, 1987, pp. 117-136.
- Kling, R., and Scacchi, W. "The Social Web of Computing: Computer Technology as Social Organization," *Advances in Computers* (21), 1982, pp. 2-90.
- Kruchten, P. *The Rational Unified Process: An Introduction*, Reading, MA: Addison-Wesley, 1998.
- Lakoff, G., and Johnson, M. *Metaphors We Live By*, Chicago: University of Chicago Press, 2003.
- Larkin, J., and Simon, H. "Why a Diagram Is (Sometimes) Worth Ten Thousand Words," *Cognitive Science* (11), 1987, pp. 65-99.
- Latour, B. "From the World of Science to that of Research?," *Science*, April 1998, pp. 14-19.
- Latour, B. *Science in Action*, Cambridge, MA: Harvard University Press, 1987.
- Latour, B. "Social Theory and the Study of Computer Work Sites," in W. J. Orlikowski, G. Walsham, M. R. Jones, and J. I. DeGross (Eds.), *Information Technology and Changes in Organizational Work*, London: Chapman & Hall, 1996, pp. 295-307.
- Latour, B. *We Never Have Been Modern (Nous n'avons jamais été modernes)*, Translated by Catherine Porter, Hemel Hempstead, England: Harvester Wheatsheaf, 1993.
- Lee, J., and Truex, D. "Exploring the Impact of Training in ISD Methods on the Cognitive Structure of Novice Information Systems Developers," *Information Systems Journal* (10:4), October 2000, pp. 347-368.
- Levine, L.; Baskerville, R.; Loveland Link, J. L.; Pries-Heje, J.; Ramesh, B.; and Slaughter, S. "Discovery Colloquium: Quality Software Development @ Internet Speed," SEI Technical Report CMU/SEI 2002-TR-020, ESC-TR-2002-020, Software Engineering Institute, Carnegie-Mellon University, 2002.
- Liu, K. *Semiotics in Information Systems Engineering*, Cambridge, England: Cambridge University Press, 2000.
- Liu, K.; Clarke, R. J.; Andersen, P. B.; and Stamper, R. K. *Organizational Semiotics: Evolving a Science of Information Systems*, Boston: Kluwer Academic Publishers, 2002a.
- Liu, K.; Sun, L.; and Bennett, K. "Co-design of Business and IT Systems—Introduction," *Information Systems Frontiers* (4:3), 2002b, pp. 251-256.
- Lyytinen, K. J., and Ngwenyama, O. K. "What Does Computer Support for Cooperative Work Mean? A Structural Analysis of Computer Supported Cooperative Work," *Accounting, Management and Information Technology* (2:1), 1992, pp. 19-37.
- Madsen, K. H. "A Guide to Metaphorical Design," *Communications of the ACM* (37:12), December 1994, pp. 57-62.
- Mathiassen, L., and Munk-Madsen, A. *Object Oriented Analysis & Design*, Aalborg, Denmark: Marko Publishers, 2000.
- Mayer, R. J.; Benjamin, P. C.; Caraway, B. E.; and Painter, M. K. "A Framework and a Suite of Methods for Business Process Reengineering," in V. Grover and W. J. Kettinger (Eds.), *Business Process Change: Reengineering Concepts, Methods, and Technologies*, Harrisburg, PA: Idea Group Publishing, 1995, pp. 245-290.
- Mylopoulos, J.; Chung, L.; and Nixon, B. "Representing and Using Non-Functional Requirements: A Process-Oriented Approach," *IEEE Transactions on Software Engineering* (18:6), June 1992, pp. 483-497.

- Nuseibeh, B. "Weaving Together Requirements and Architectures," *IEEE Computer* (34:3), 2001, pp. 115-117.
- Nuseibeh, B., and Easterbrook, S. "Requirements Engineering: A Roadmap," *Software Engineering* (20:10), 2000, pp. 760-773.
- Nuseibeh, B.; Kramer, J.; and Finkelstein, A. C. W. "A Framework for Expressing the Relationships between Multiple Views in Requirements Specification," *IEEE Transactions on Software Engineering* (20:10), 1994, pp. 760-773.
- OMG. "Unified Modeling Language, Standard Specification Version 2.0," Object Management Group, 2003 (available online at <http://www.omg.org/uml>, accessed November 30, 2003).
- Opdahl, A. L.; Henderson-Sellers, B.; and Barbier, F. "An Ontological Evaluation of the OML Metamodel," in E. D. Falkenberg, K. Lyytinen, and A. A. Verrijn-Stuart (Eds.), *Information Systems Concepts: An Integrated Discipline Emerging*, Boston: Kluwer Academic Publishers, 2000, pp. 217-232.
- Orlikowski, W. J. "The Duality of Technology: Rethinking the Concept of Technology in Organizations," *Organization Science* (3:3), 1992, pp. 398-429.
- Orlikowski, W. J. "Improvising Organizational Transformation Over Time: A Situated Change Perspective," *Information Systems Research* (7:1), 1996, pp. 63-92.
- Orlikowski, J. "Using Technology and Constituting Structures: A Practice Lens for Studying Technology in Organizations," *Organization Science* (11:4), July-August, 2000, pp. 404-428.
- Orlikowski, W. J., and Iacono, C. S. "Research Commentary: Desperately Seeking the 'IT' in IT Research—A Call to Theorizing the IT Artifact," *Information Systems Research* (12:2), 2001, pp. 121-134.
- Orlikowski, W. J., and Robey, D. "Information Technology and the Structuring of Organizations," *Information Systems Research* (2:2), June 1991, pp. 143-169.
- Parsons, J., and Wand, Y. "Emancipating Instances from the Tyranny of Classes in Information Modeling," *ACM Transactions on Database Systems* (25:2), 2000, pp. 228-268.
- Patel, N. V. "Developing Tailorable Information Systems through Deferred System's Design," in W. D. Haseman and D. L. Nazareth (Eds.), *Proceedings of the 5th Americas Conference on Information Systems*, Milwaukee, WI, 1999, pp. 4-6.
- Pressman, R. S., and Associates. "Glossary of Software Engineering Terms," 2003 (available online at <http://www.rspa.com/spi/glossary.html>; accessed September 30, 2003).
- Probert, S. K., and Rogers, A. "Understanding Hard and Soft IS Development Methods: Paradigmatic Rigidities or Different Ends of a Spectrum?," in W. D. Haseman and D. L. Nazareth (Eds.), *Proceedings of the 5th Americas Conference on Information Systems*, Milwaukee, WI, 1999, pp. 650-652.
- Purao, S.; Rossi, M.; and Bush, A. "Towards an Understanding of the Use of Problem and Design Spaces During Object-Oriented System Development," *Information and Organization* (12:4), 2002, pp. 249-281.
- Purao, S.; Truex, D.; and Cao, L. "Now the Twain Shall Meet: Combining Social Sciences and Software Engineering to Support Development of Emergent Systems," in G. Ross and D. Galletta (Eds.), *Proceedings of the 9th Americas Conference on Information Systems*, Tampa, FL, 2003, pp. 2738-2744.
- Ramesh, B., and Jarke, M. "Towards Reference Models for Requirements Traceability," *IEEE Transactions on Software Engineering* (27:1), 2001, pp. 58-93.
- Robinson, W. N.; Pawlowski, S. D.; and Volkov, V. "Requirements Interaction Management," *ACM Computing Surveys* (35:2), 2003, pp. 132-190.
- Rossi, M., and Brinkkemper, S. "Complexity Metrics for Systems Development Methods and Techniques," *Information Systems Journal* (21:2), 1996, pp. 209-227.
- Rossi, M.; Gustafsson, M.; Smolander, K.; Johansson, L.-A.; and Lyytinen, K. "Metamodeling Editor as a Front End Tool for a CASE Shell," in P. Loucopoulos (Ed.), *Advanced Systems Engineering, CAiSE '92*, Manchester, UK, May 12-15, 1992, pp. 546-567.

- Schwaber, K., and Beedle, M. *Agile Software Development with Scrum*, Upper Saddle River, NJ: Prentice Hall, 2002.
- Siau, K.; Wand, Y.; and Benbasat, I. "The Relative Importance of Structural Constraints and Surface Semantics in Information Modeling," *Information Systems Journal* (22:2/3), 1997, pp. 155-170.
- Stamper, R. K. "Information Systems as a Social Science: An Alternative to the FRISCO Formalism," in E. D. Falkenberg, K. Lyytinen, and A. A. Verrijn-Stuart (Eds.), *Information Systems Concepts: An Integrated Discipline Emerging*, Boston: Kluwer Academic Publishers, 2000, pp. 1-51.
- Stamper, R. "Social Norms in Requirements Analysis—An Outline of MEASUR," in M. Jirotko and J. A. Goguen (Eds.), *Requirements Engineering: Technical and Social Aspects*, New York: Academic Press, 1994, pp. 107-139.
- Suchman, L. A. "What is Human-Machine Interaction," in S. P. Robertson, W. Zachary, and J. B. Black (Eds.), *Cognition, Computing, and Cooperation*, Norwood, NJ: Ablex Publishing Corporation, 1990, pp. 25-55.
- Truex, D. P.; Baskerville, R.; and Klein, H. K. "Growing Systems in an Emergent Organization," *Communications of the ACM* (42:8), August 1999, pp. 117-123.
- Truex, D. P.; Baskerville, R.; and Travis, J. "Amethodical Systems Development: The Deferred Meaning of Systems Development Methods," *Accounting Management and Information Technologies* (10), 2000, pp. 53-79.
- Van der Aalst, W. M. P.; ter Hofstede, A. H. M.; and Weske, M. "Business Process Management: A Survey," in W. M. P. van der Aalst, A.H.M. ter Hofstede, and M. Weske (Eds.), *Proceedings of the International Conference on Business Process Management*, 2003, pp. 1-12.
- Walsham, G. "Cross-Cultural Software Production and Use: A Structural Analysis," *MIS Quarterly* (26:4), 2002, pp. 359-379.
- Walsham, G., and Han, C. K. "Structuration Theory and Information Systems Research," *Journal of Applied Systems Analysis* (17), 1991, pp. 77-85.
- Wand, Y., and Weber, R. "On the Ontological Expressiveness of Information Systems Analysis and Design Grammars," *Journal of Information Systems* (3:4), 1993, pp. 217-237.
- Wand, Y., and Weber, R. "Research Commentary: Information Systems and Conceptual Modeling—A Research Agenda," *Information Systems Research* (13:4), 2002, pp. 363-376.
- Weber, R. "Editor's Comments: Still Desperately Seeking the IT Artifact," *MIS Quarterly* (27:2), June 2003a, pp. iii-xi.
- Weber, R. "Editor's Comments: Theoretically Speaking," *MIS Quarterly* (27:3), September 2003b, pp. iii-xiii.
- Welke, R. J. "A Context Approach to Information Systems Development Synthesis," ISRAM Working Paper W-8007-1.0, McMaster University, Hamilton, Ontario, Canada, 1980.
- Wieringa, R. "A Survey of Structured and Object-Oriented Software Specification Methods and Techniques," *ACM Computing Surveys* (30:4), 1998, pp. 459-527.
- Zave, P. "Classification of Research Efforts in Requirements Engineering," *ACM Computing Surveys* (29:4), 1997, pp. 315-321.

ABOUT THE AUTHORS

Sandeep Purao received his Ph.D. in management information systems from the University of Wisconsin-Milwaukee in May 1995. His research focuses on various aspects of information system design and development in organizational settings. His current projects include reuse-based design of information systems, evaluation of information systems, co-design of information systems and organizations, and measurement of information systems. His work has been pub-

lished in *Information Systems Research*, *ACM Computing Surveys*, *Communications of the ACM*, *IEEE Transactions on Systems, Man and Cybernetics*, *Information & Organizations*, *Decision Support Systems*, and other venues.

Duane Truex researches the social impacts of information systems and emergent ISD. He is an associate editor for the *Information Systems Journal*, has coedited two special issues of *The Database for Advances in Information Systems*, and is on the editorial board of the *Scandinavian Journal of Information Systems*, the *Journal of Communication*, *Information Technology & Work*, and the *Online Journal of International Case Analysis*. His work has been published in *Communications of the ACM*, *Accounting Management and Information Technologies*, *The Database for Advances in Information Systems*, *European Journal of Information Systems*, *le journal de la Soci t  d'Information et Management*, *Information Systems Journal*, *Journal of Arts Management and Law*, *IEEE Transactions on Engineering Management*, and 40 assorted IFIP transactions, edited books, and conference proceedings. He is a member of the Decision Sciences and Information Systems faculty in the Chapman Graduate School, College of Business, at Florida International University, and is an associate professor on leave from the Computer Information Systems Department, Robinson College of Business, at Georgia State University.