12-31-2003

Now the Twain Shall Meet: Combining Social Sciences and Software Engineering to Support Development of Emergent Systems

Sandeep Purao
Pennsylvania State University

Duane Truex
Florida International University, dtruex@gsu.edu

Lan Cao
Georgia State University

NOW THE TWAIN SHALL MEET: COMBINING SOCIAL SCIENCES AND SOFTWARE ENGINEERING TO SUPPORT DEVELOPMENT OF EMERGENT SYSTEMS

Sandeep Purao
Pennsylvania State University
spurao@ist.psu.edu

Duane Truex
Florida International University
Duane.Truex@fiu.edu

Lan Cao
Georgia State University
lcao@gsu.edu

Abstract

This paper introduces a research program aimed at aligning two research communities which share similar aims and research questions, but which are largely unaware of the work and the epistemological stance of the other. Both communities are concerned with the design and deployment of information systems in organizations and both recognize the mutual interdependence of these technologies and social systems. But each of these communities struggles with the nature of those interactions in different ways. It is our contention that by calling attention to the differences and similarities in approaches we maybe able to tap into a useful synergy. We hope to further the development of mid-range theories and models to help align these two epistemological stances.

Keywords: Information system development, problem space, design space, representation techniques

Introduction

Much current research in information system development (ISD) focuses on creating IT artifacts (Information Systems) that support users' activities in organizational contexts. The software engineering community, in focusing on techniques and processes for developing these artifacts, has paid insufficient attention to the complex relationship between the Information System (IS) and the organizational context in which it is embedded (Orlikowski and Iacono 2001). On the other hand, branches of the MIS community, such as the IFIP Working Group 8.2, have paid limited attention to translating their insights on organizational context into actionable techniques that may be used for building more effective information systems. Without the benefit of integration, these two streams have traveled down paths that may be viewed as increasingly specialized, making dialog difficult between the two communities. We suspect that this lack of dialog has resulted in increasing emphasis on minutia and decreasing relevance of the research outcomes to the IS community. As an applied research discipline, IS research is concerned with the understanding of IT in organizations and society. Without the benefit of an integrated perspective, the research outcomes, therefore, are likely to touch upon only parts of the solutions required for solving complex problems.

This paper makes an attempt to combine important insights from these two research streams. A specific goal of the paper is to argue for the development of specific representational techniques that achieve this integration. The remainder of the paper is organized in three sections. Section 2 outlines theories that have driven investigation of IT in organizations. In section 3, we describe current research thrusts in software engineering. The reconciling of the two research streams is discussed in section 4. Finally, section 5 develops the argument for representational techniques that can help operationalize the lessons learned from social theories with the help of tools available to software engineers.

Social Science Theories to Understand IS/IT in Organizations

Understanding the relationship between technology and society is at the heart of IS research tradition. Social theories try to understand, explain and predict social behavior. They have long been used to help explain the underlying logics of social organizations. In the IS Research discipline the IFIP WG 8.2 research community[1] provides a microcosm of the social theory in IS debate. In its publications one can find a very extensive use of and debate about the efficacy of various social theories. In a recent analysis of social theory citations in the IFIP WG8.2 proceedings, Jones (Jones 2000) showed how the use of different social theories has evolved within the WG 8.2 community since its inception in 1977. We use this research community and Jones's research as a surrogate of the whole of our research community to illustrate our point. He performed a content analysis on the working group proceedings identifying the predominant theories each paper in those proceedings. The following figure illustrates relative frequencies of the major social theories employed in these works.

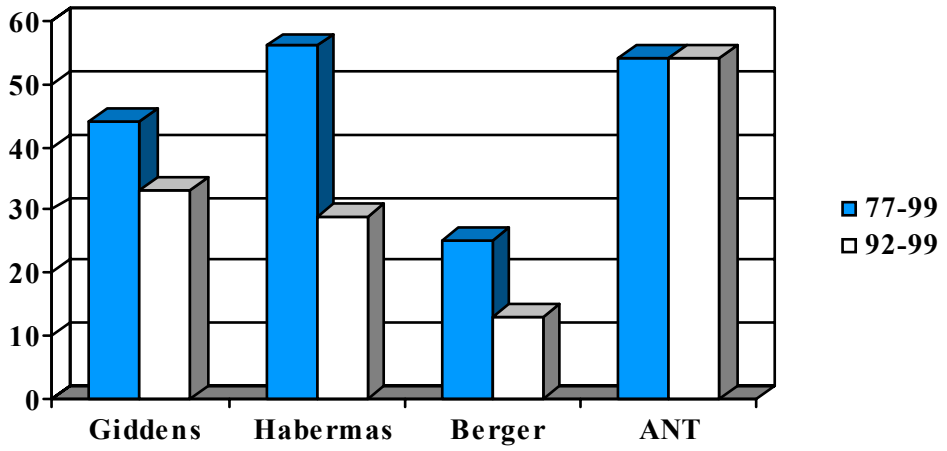

Figure 1. Social Theorists Cited in IFIP WG 8.2 (Adapted from Jones, 2000)

One theoretical perspective that found particularly relevant to research in ISD is Giddens' Structuration Theory, which can be used to capture the complex relationship between the IS and its organizational context (Giddens 1979, 1984, 1987, 1990; Giddens and Turner 1987). While the application of structuration theory to IS research continues to be a debatable issue, most would agree that the theory is well suited for exploring the ephemeral boundaries between IT and organization. Several authors have used it to explore how IT's dual characters as enablers and constrain of social action. Thus adaptations of structuration theory have proved useful to study various aspects of information technology in organizations (Barley 1986a; 1986b; DeSanctis and Poole 1994; Orlikowski and Robey 1991; Orlikowski 1992, 1996, 2000; Orlikowski, et al. 1996; Rose and Lewis 2000). A key principle in structuration theory is the reciprocal nature of structure: human action is enabled and constrained by structure, but structure is also the result of human action. Thus, the duality in structuration theory refers to the way in which action and structure

[1]The International Federation of Information Processing (IFIP) has a mission to "to be the leading, truly international, apolitical organization which encourages and assists in the development, exploitation and application of Information Technology for the benefit of all people." [IFIP 2003]. IFIP can trace its roots to the first major international conference on computers and computing held in Paris in 1959 under the auspices of UNESCO and was founded to foster international cooperation, to stimulate research, development and applications and to encourage education and the dissemination and exchange of information on all aspects of computing and communication.

Its work is accomplished through its 12 technical committees and 83 working groups whose membership includes thousands of information scientists world wide. These groups cooperate and foster knowledge growth and cooperation via sponsorship of open conferences, smaller working conferences, seminars and tutorials, circulated papers, electronic conferencing and other electronic interaction. Technical committees 8 (TC8) (Information Systems) and TC9 (Relationship between Computers and Society) have 15 working groups addressing various aspects of information technologies in social settings. In particular the focus of IFIP WG 8.2, is to address the Interaction of Information Systems and the Organization. It has organized 19 working conferences, and published proceedings with more than 370 papers discussing all manner of social theories as they apply to IS development, implementation and use in organizations. In its 20-year history the working group is widely acknowledged to have helped spearhead the qualitative turn in IS research and has provided a sounding board for reflection upon the interaction of organizations and information technology.

presupposes each other. That is, social structure provides enabling and constraining elements that are drawn on in human interactions, and in so doing social structures are produced and reproduced.

Other social theories address related questions, but from different points of view. Critical Social Theory as applied to IS research addresses issues of power differentials in ISD and IS use. The works of Habermas, Bourdieu and Apel (Apel 1980; Bourdieu 1991; Habermas 1981, 1984) have motivated this work in our field. Actor Network Theory addresses the complex relationships between human and non-human actants in rich social and technological domains. The works of Latour (1987, 1993, 1996, 1998) and by Callon (1986, 1991) have informed this to IS research stream. Other authors have explored the question of what is the most beneficial way to appropriate and integrate social theories into our research (Holmström and Truex 2001). And, of course, we could choose other theories which have proved important as well. But for the purposes of this paper we choose the limited set to illustrate a point of intersection and departure in the research in these two research communities.

A Software Engineering Perspective on IS/IT in Organizations

As the name suggests, the software engineering perspective considers IS/IT as an artifact to be 'engineered,' that is, developed, built, and deployed. The primary focus, therefore, is on techniques and models that will allow design and production of software artifacts. The organization, then, is often seen as the source of requirements, which must be converted to formats appropriate for realization in the software artifact. Much of research in software engineering takes a dualistic approach, in seeing the two domains of the IS artifact and the organization as being separate rather than a dualism in which the two are mutually joined in a dance of perpetual structuring, each influencing the other – a perspective repeatedly put forward by the social theorists. Few software engineering methods, tools and techniques have paid explicit limited attention to this interaction between IS and organizations. Many methods and the concomitant modeling techniques assume relative stability in organizational problem domains and proceed to quickly move to developing generic organizational use case, or process instances.

Research within ISD does, however, suggest that the problem domain (organizational context) is intertwined with the design domain (the IS artifacts), and may change with the opportunity or need to redesign the business process (Mathiassen and Stage 1992; Welke 1994). Bergman et al. (2001), for example, point out that ISD process creates goals that require adoption of both the business process as well as design of the information system. This interplay is captured by the ideas of 'Problem' and 'Design' spaces, which represent a key dimension that underlies ISD processes (Guindon 1990; Mathiassen and A. Munk-Madsen 2000; Purao 2002). Purao et al (2002) employ grounded theory development techniques to present evidence of the existence of these two spaces, and describe patterns of behaviors across the two spaces. Their study, accordingly, laments the lack of supporting techniques to bridge the gap between problem and design spaces – mirroring the larger concern voiced by Mathiassen and Stage (1992)and Bergman et al. (2001) and suggesting possible reasons for documented problems with the failures in Information Systems development and acceptance (Dardenne, et al. 1993).

In spite of this recognition of interplay and dynamics across problem and design spaces, a key goal of current ISD practice and research has been the minimization of change, either with better articulation of requirements or flexible design, i.e. assuming either an unchanging problem space or adjusting the design space in response to changes in the problem space. Accordingly, several modeling techniques have been proposed to 'deal with complexity,' 'allow for flexibility,' and 'facilitate communication between developers and users.' As a result, many modeling and representation techniques have been developed to capture different perspectives of the underlying IT artifact – including functionality, structure, behavior, and dynamics. Developing an information system, however, requires understanding of problem space – e.g. domain knowledge which is considered informal, implicit, ad hoc, and modeled only incompletely and indirectly in terms of problem-specific languages (Iscoe, et al. 1991). Only a few techniques have been proposed to model the problem domain such as the use of flowcharts and dataflow diagrams to model the current functions (Gane and Sarson 1979), the IDEF suite of process maps (Mayer, et al. 1995). These have found little acceptance in practice, and have generally been considered too complicated for users to understand (Dawson and Swatman 1999). Further, they have not directly addressed concepts such as organizational actors, activities and tasks; instead, focusing on only those aspects of the organizational context that will eventually be represented in an information system. Another technique is rich pictures (Avison and Wood-Harper 1990), which are cartoon-like representations that identify all the stakeholders, their concerns, and some of the structure underlying the work context. This technique, while easy to use, has suffered from lack of integration with more formalized approaches.

On the other hand, in design space a large number of methods have been proposed for the specification of software systems in the past 20 years. Until 1988, the proposed methods followed the structured approach and since that time, most proposals have followed the object-oriented approach. A survey by Wieringa (1998)classified the techniques as those modeling the functional,

behavioral, and communicational properties of a system as well as decomposition techniques. Examples of these techniques include: function refinement trees, event-response specification, uses cases, process graph, process structure diagram, state transition diagram, state charts, Data Flow Diagrams (DFD), sequence diagrams, collaboration diagrams, context diagrams, Entity-Relationship Diagrams (ERD) and class diagrams. In terms of the framework proposed by Wand and Weber (2002), research on these techniques has concentrated on conceptual modeling grammars, methods, scripts and context. There are few techniques, if any, that have dealt with the important concern of representing context, and further, influencing context. These modeling techniques have, therefore, continued to focus on the design space (see Figure 2)

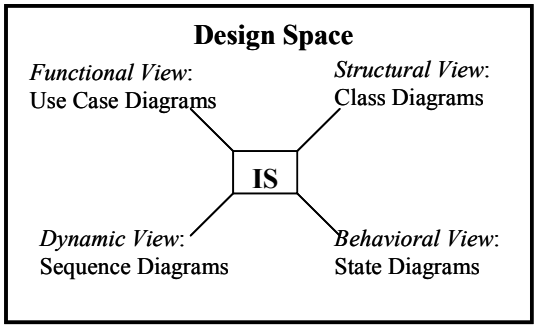

Figure 2. Techniques for Modeling the Design Space

ISD as Co-Evolution of Problem and Design Space

Both streams have been slow at recognizing that the uni-dimensional views embraced within the stream may be inadequate to deal with the difficult problems associated with developing complex information systems. ISD process is a dynamic process that needs to capture the changing requirements and constraints. Change management is a fundamental activity in requirement engineering (Nuseibeh and Easterbrook 2000). To deal with the interaction with the changing business environment, prior research in ISD has focused on improving software development process. For example, spiral model (Boehm 1988) describes software development as the iteration over four phases of activity. The specification and plan are refined at each iteration. Another mechanism to manage the changing environment is to trace changes to requirements, design and implementation of the system. With traceability, requirements are linked to their sources and to the artifacts created during the ISD cycle. Therefore, the impact of change in requirements can be identified (Ramesh 2001). Bergman et al (2001) have suggested that requirements analysis should be seen as an iterative process from an existing solution space to problem space, then to the future solution space. The two spaces are "grinding" as each provides input to the other and each is adjusted to reflect the decisions in the other. One might think of this as being a dialectical process where the interaction of competing worldviews force a reconsideration and refinement of each space. The process is a kind of hermeneutic cycle. While their characterization of problem and solution space is somewhat different from the characterization in Purao et al.(2000), their idea of heterogeneous engineering does bring out anomalies in existing business process not supported by the current systems as the problem space. Their metaphor of 'grinding' to capture this dynamic nature of problems and solutions, and their continuous discovery and simultaneous interpretation/creation (see Figure 3) is, however, clearly relevant to our characterization of problem and design spaces.

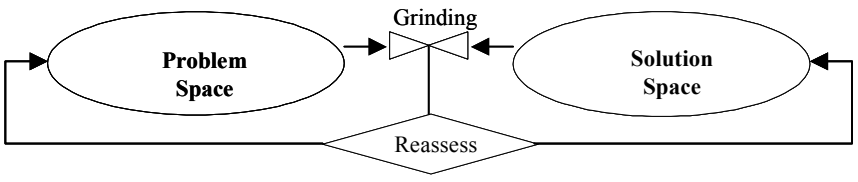

Figure 3. Heterogeneous Engineering [Adapted from Bergman et al. [2001]]

As the discussion above suggests, the software engineering stream is beginning to realize that the exclusive focus on the design space (the IT artifacts) overlooks the possible intervention that a system developer may be able to make in the problem space (the organization). Problem space and design space represent two distinct arenas of influence for ISD. The problem space is a subset

of the real world that the computer-based information system is designed for. It describes the business processes, roles involved in the processes and distribution of power and resources. Problems of current system and user requirements of a system are analyzed based on understanding the problem space. The design space is the solution of the problems identified in the problem space and the design of the system itself. The idea has been alluded to and received attention over the years (Guindon 1990, 1986; Mathiassen and A. Munk-Madsen 2000; Simon 1996). Developer behaviors in the problem space differ considerably from those in the design space (Purao et al. 2002). For example, in problem space, developers engage in concept simulation in an effort to explore different concepts and visualizing a mental representation of concepts. In design space developers engage in behaviors such as expanding mental models, proposing new concepts, considering action sequence among those new concepts and visualizing how the artifact may behave. Successfully engaging in each space is critical to the quality of the system. While there are tools and techniques supporting behaviors in design space, very few techniques or procedures are available in supporting developer behaviors in the problem space (Purao et al. 2002). If the ISD process is recast as simultaneously causing changes in both spaces, it can be seen as proactive in modeling, simulating and evaluating the changes and the effects of the changes on both the organization and the information system being developed (Purao and Cao 2003).

The social theories stream is also slowly shifting its agenda, borrowing from the theory of linguistic and organizational emergence, to the line or work loosely called "emergent systems development" or "deferred systems development" (Bansler, et al. 2000; Bansler and Havn 2002; Bello, et al. 2002; Patel 1999; Patel, et al. 2002; Truex, et al. 1999; Truex, et al. 2000). Deferred system's design (DSD) provides some elements to consider in designing context-aware IS. Most context-aware systems record information on users' actions for later use. In IS this is reflected in work that seeks to predict users' future requirements. The IS design is based on predicting what users will require in order to implement adaptation. The DSD approach does not seek to predict specific future uses of an IS or develop designs on a predictable basis. Rather, it proposes design principles that can be used to develop a basic research agenda on the concept of deferred design. To adapt these into deferred systems development settings through real-time tailoring of organizational environment and context. *A deferred system design is defers fixing or freezing the system until the user decides what the system will become* (Patel 1999, 2002). Of course this implies that if the target organizational setting is emergent and continually *in process*, or in more philosophical terms, *becoming,* versus being complete or finished, then the system will be a kind of perpetual prototype. In business terms, it caters for the situated needs of system users. Emergent systems development views organizations as emergent and then illustrates how there should be a consequent shift from traditional development practices to a "continuous redevelopment" process (Truex, et al. 1999).

A Research Agenda

Representational techniques are needed to bridge problem and design spaces. A representational technique is important because it brings together a community of researchers by focusing attention on key aspects that can be abstracted in the form of modeling constructs. It allows the research community to come to an agreement about key constructs within a discipline. The representations may come in different forms, some more concrete and others conceptual. On the one hand, the Design Research community has tried to make the conceptual problem space of the organization as concrete as possible and to move as rapidly as feasible to a system model and specification that may actually be realized in code and in hardware. The more theoretical ISD community typified by IFIP WG 8.2, on the other hand, has been more comfortable with allowing a higher level of model abstraction and has been less concerned with the "nitty gritty" of building system instances. But both communities are unfulfilled to the extent that they do not effectively address one domain as well as the other.

A good example, and perhaps an analogy, of how a research community can get organized in this manner is the entity-relationship diagramming technique proposed by Chen (1976). The simple yet elegant proposal allowed the research community to begin thinking about research in databases in the form of a metaphor that captured the essence of databases as reflecting a universe of IS course. More recently, the agenda by Wand and Weber (2002) has presented such an opportunity. Our approach represents a call for extending that agenda to include the problem space.

The problem of developing techniques for modeling the problem space, and for bridging models across the two spaces is a difficult one. One of the difficult problems in modeling the problem space is choice of an adequate representation. The information obtained in problem space, particularly at the early stages, is usually informal, subject to rapid change, contains different personal views, and demonstrates little system understanding (Jarke and Pohl 1993). Developers then move to semi-formal graphics, and in some situations also to formal specifications (Lubars, et al. 1993).

To support such behaviors at different stages of a project, we may need a two-layer approach to represent problem space: the first layer is a textual representation layer that retains the basic functionality of a standard text-processing user interface while adding

traceability to constructs in the second layer. The second layer—a graphical model layer is a more structured representation of the problem space. There are of course other as yet unexplored options and approaches. We invite other researchers to join in the search for ways to bridge the distance.

References

Apel, K. *Towards a Transformation of Philosophy*, Routledge & Kegan Paul, London, 1980.

Avison, D.E., and Wood-Harper, A.T. *Multiview: An Exploration in Information Systems Development*, Blackwell Scientific Publications, Oxford, 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), 2000.

Bansler, J.P., and Havn, E. "Improvisation and Bricolage in Information Systems Development: A Field Study," Copenhagen, 2002, pp. 10.

Barley, S.R. (ed.). *The Social Construction of a Machine: Ritual, Superstition, Magical Thinking and Other Pragmatic Responses to Running a Ct Scanner*, Riedel Press, 1986a.

Barley, S.R. "Technology as an Occasion for Structuring: Evidence from Observations of Ct Scanners and the Social Order of Radiology Departments," *Administrative Science Quarterly* (31), 1986b, pp. 78-108.

Bello, M., Sorrentino, M., and Virili, F. "Web Services and Emergent Organizations: Opportunities and Challenges for Is Development," *Proceedings of the ECIS 2002 Information Systems and the Future for the Digital Economy: Xth European Conference on Information Systems*, Gdansk, Poland, 2002.

Bergman, M., King, J., and Lyytinen, K. *Large Scale Requirements Analysis as Heterogeneous Engineering. In Social Thinking - Software Practice*, MIT Press, Cambridge, MA, 2001.

Boehm, B. "A Spiral Model of Software Development and Enhancement," *Computer* (11:4), 1988.

Bourdieu, P. *Language and Symbolic Power*, Harvard University Press, Cambridge, 1991.

Callon, M. "Some Elements of a Sociology of Translation: Domesticationof the Scallops and the Fishermen of St Brieuc Bay," In *Power Action and Belief: A New Sociology of Knowledge*, J. Law (ed.) 32, Routledge, London, 1986.

Callon, M. "Techno-Economic Networks and Irreversibility," In *A Sociology of Monsters: Essays on Power, Technology and Domination*, J. Law (ed.) Routledge, London, 1991, pp. 132-161.

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.

Dardenne, A., Lamseerde, A.v., and Fickas, S. "Goal-Directed Requirements Acquisition," *Science of Computer Programming* (20), 1993, pp. 3-50.

Dawson, L., and Swatman, P. "The Use of Object-Oriented Models in Requirements Engineering: A Field Study," *Proceedings of the Proceeding of the 20th international conference on Information Systems*, Charlotte, North Carolina, 1999.

DeSanctis, G., and Poole, M.S. "Capturing the Complexity in Advanced Technology Use: Adaptive Structuration Theory," *Organization Science* (5:2), 1994.

Gane, E.C., and Sarson, T. *Structured Systems Analysis: Tools and Techniques*, Prentice-Hall, Englewood Cliffs, NJ, 1979.

Giddens, A. *Central Problems in Social Theory: Action, Structure and Contradiction in Social Analysis.*, University of California Press, Berkeley, Calif., 1979.

Giddens, A. *The Constitution of Society: Outline of the Theory of Structuration*, Polity Press, Cambridge, 1984.

Giddens, A. *A Contemporary Critique of Historical Materialism*, Berkeley University Press., Berkeley, CA., 1987.

Giddens, A. *The Consequences of Modernity*, Polity Press, Cambridge, 1990.

Giddens, A., and Turner, J.H. *Social Theory Today*, Polity Press, Cambridge, 1987.

Guindon, R. "Designing the Design Process: Exploiting Opportunistic Thoughts," *Human-Computer Interaction* (5), 1990.

Guindon, R., H. Krasner, and B. Curtis, *Breakdowns and Processes During the Early Activities of Software Design by Professionals*, 1986.

Habermas, J. "Reason and the Rationalization of Society," In *The Theory of Communicative Action*, One, Beacon Press, Boston, 1981, pp. 325.

Habermas, J. *The Theory of Communicative Action*, Beacon Press, Boston, 1984.

Holmström, J., and Truex, D.P. "What Does It Mean to Be an Informed Is Researcher? Some Criteria for the Selection and Use of Social Theories in Is Research," *Proceedings of the IRIS 24 (Information Systems Research in Scandinavia 2001)*, Bergen, Norway, 2001, pp. 313-326.

IFIP 2003. "IFIP at your fingertips," **http://www.ifip.or.at/**, Access Date March 15, 2003.

Iscoe, N., Williams, G.B., and Arango, G. "Domain Modeling for Software Engineering," *Proceedings of the Proceedings of the 13th International Conference on Software Engineering*, Austin, TX., 1991, pp. 340-343.

Jarke, M., and Pohl, K. "Establishing Visions in Context: Toward a Model of Requirements Process," *Proceedings of the ICIS 93*, Orlando, FL, 1993.

Jones, M.R. "The Moving Finger: The Use of Social Theory in Wg8.2 Conference Papers, 1975-1999," In *Organizational and Social Perspectives on Information Technology*, R. Baskerville, J. Stage and J. I. DeGross (eds.), Kluwer Academic Publishers, Dordrecht, 2000, pp. 15-31.

Latour, B. *Science in Action*, Harvard University Press, Cambridge, Mass., 1987.

Latour, B. *We Never Have Been Modern (Nous N'avons Jamain Été Modernes)*, Harvester Wheatsheaf, Hemel Hempstead, 1993.

Latour, B. "Social Theory and the Study of Computer Work Sites," In *Information Technology and Changes in Organizational Work*, W. J. Orlikowski, G. Walsham, M. Jones and J. I. DeGross (eds.), Chapman and Hall, London, 1996, pp. 295-307.

Latour, B. "From the World of Science to That of Research?," *Science*:April 1998), 1998, pp. 14-19.

Lubars, M., Potts, C., and Richter, C. "A Review of the State of the Practice in Requirements Modeling," *Proceedings of the IEEE Symposium on Requirements Engineering (RE'93)*, San Diego, 1993.

Mathiassen, L., and Munk-Madsen, A. *Object Oriented Analysis and Design*, Marko Publishers, Aalborg, Denmark, 2000.

Mathiassen, L., and Stage, J. "The Principle of Limited Reduction in Software Design," *Information Technology and People* (6:2-3), 1992.

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 *Business Process Change: Reengineering Concepts, Methods, and Technologies*, V. Grover and W. J. Kettinger (ed.) Idea Group Publishing, Harrisburg, PA, 1995, pp. 245-290.

Nuseibeh, B., and Easterbrook, S. "Requirements Engineering: A Roadmap," *Proceedings of the Proceedings of the conference on the future of Software engineering*, 2000.

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., and Robey, D. "Information Technology and the Structuring of Organizations," *Information Systems Research* (2:2), 1991, pp. 143-169.

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, W.J., Walsham, G., and Jones, M. "Information Technology and Changes in Organizational Work: Images and Reflections," In *Information Technology and Changes in Organizational Work*, W. J. Orlikowski, G. Walsham, M. Jones and J. I. DeGross (eds.), Chapman and Hall, London, 1996, pp. 1-10.

Orlikowski, W. J. and Iacono, S. "Desperately Seeking the "IT" in IT Research-A Call to Theorizing the IT Artifact," *Information Systems Research*, 12(2), 2002, pp. 121-134.

Patel, N.V. "Developing Tailorable Information Systems through Deferred System's Design," *Proceedings of the Americas Conference on Information Systems*, Milwauikee, WI, August 13-15, 1999.

Patel, N.V., Dittrich, Y., Eardley, A., and Lycett, M. "Deferred System's Design: Developing Context-Aware Information Systems for Dynamic Environments (a Panel Discussion)," *Proceedings of the ECIS 2002 Information Systems and the Future for the Digital Economy: Xth European Conference on Information Systems*, Gdansk, Poland, June 6--8, 2002, 2002.

Purao, S., and Cao, L. "Research in Isd: Supporting Co-Evolution of Information Systems and Organizations," *Proceedings of the Symposium on Systems Analysis and Design*, Miami, Fl, 2003.

Purao, S., M. Rossi and A. Bush, "Towards an Understanding of the Use of Problem and Design Spaces During Object-Oriented System Development," *Information and Organization* (12:4), 2002.

Ramesh, B.a.M.J. "Towards Reference Models for Requirements Traceability," *IEEE Transactions on Software Engineering* (37:1), 2001.

Rose, J., and Lewis, P. "Using Structuration Theory in Action Research: An Intranet Development Project," In *Realigning Research and Practice in Information Systems Development: The Social and Organizational Perspective,*, J. L. DeGross (ed.) Kluwer Academic Publishers, Boston, 2000, pp. 273-296.

Simon, H.A. *The Sciences of the Artificial*, Cambridge: The MIT Press, 1996.

Truex, D.P., Baskerville, R., and Klein, H.K. "Growing Systems in an Emergent Organization," *Communications of the ACM* (42 no. 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.

Wand, Y., and Weber, R. "Research Commentary: Information Systems and Conceptual Modeling -- a Research Agenda," *Information Systems Research* (13:4), 2002, pp. 363-376.

Welke, R. "The Shifting Software Development Paradigm," *DATA BASE* (25:4), 1994.

Wieringa, R. "A Survey of Structured and Object-Oriented Software Specification Methods and Techniques," *ACM Computing Surveys* (30:4), 1998.