

Appendix A

MIS Quarterly

RESEARCH ESSAY

DESIGN SCIENCE IN INFORMATION SYSTEMS RESEARCH¹

By: Alan R. Hevner
Information Systems and Decision
Sciences
College of Business Administration
University of South Florida
Tampa, FL 33620
U.S.A.
ahevner@coba.usf.edu

Salvatore T. March
Owen Graduate School of Management
Vanderbilt University
Nashville, TN 37203
U.S.A.
Sal.March@owen.vanderbilt.edu

Jinsoo Park
College of Business Administration
Korea University
Seoul, 136-701
KOREA
jinsoo.park@acm.org

Sudha Ram
Management Information Systems
 Eller College of Business and Public
Administration
University of Arizona
Tucson, AZ 85721
U.S.A.
ram@bpa.arizona.edu

Abstract

Two paradigms characterize much of the research in the Information Systems discipline: behavioral science and design science. The behavioral-science paradigm seeks to develop and verify theories that explain or predict human or organizational behavior. The design-science paradigm seeks to extend the boundaries of human and organizational capabilities by creating new and innovative artifacts. Both paradigms are foundational to the IS discipline, positioned as it is at the confluence of people, organizations, and technology. Our objective is to describe the performance of design-science research in Information Systems via a concise conceptual framework and clear guidelines for understanding, executing, and evaluating the research. In the design-science paradigm, knowledge and understanding of a problem domain and its solution are achieved in the building and application of the designed artifact. Three recent exemplars in the research literature are used to demonstrate the application

¹Allen S. Lee was the accepting senior editor for this paper.

of these guidelines. We conclude with an analysis of the challenges of performing high-quality design-science research in the context of the broader IS community.

Keywords: Information Systems research methodologies, design science, design artifact, business environment, technology infrastructure, search strategies, experimental methods, creativity

Introduction

Information systems are implemented within an organization for the purpose of improving the effectiveness and efficiency of that organization. Capabilities of the information system and characteristics of the organization, its work systems, its people, and its development and implementation methodologies together determine the extent to which that purpose is achieved (Silver et al. 1995). It is incumbent upon researchers in the Information Systems (IS) discipline to “further knowledge that aids in the productive application of information technology to human organizations and their management” (ISR 2002, inside front cover) and to develop and communicate “knowledge concerning both the management of information technology and the use of information technology for managerial and organizational purposes” (Zmud 1997).

We argue that acquiring such knowledge involves two complementary but distinct paradigms, behavioral science and design science (March and Smith 1995). The behavioral-science paradigm has its roots in natural science research methods. It seeks to develop and justify theories (i.e., principles and laws) that explain or predict organizational and human phenomena surrounding the analysis, design, implementation, management, and use of information systems. Such theories ultimately inform researchers and practitioners of the interactions among people, technology, and organizations that must be managed if an information system is to achieve its stated purpose, namely improving the effective-

ness and efficiency of an organization. These theories impact and are impacted by design decisions made with respect to the system development methodology used and the functional capabilities, information contents, and human interfaces implemented within the information system.

The design-science paradigm has its roots in engineering and the sciences of the artificial (Simon 1996). It is fundamentally a problem-solving paradigm. It seeks to create innovations that define the ideas, practices, technical capabilities, and products through which the analysis, design, implementation, management, and use of information systems can be effectively and efficiently accomplished (Denning 1997; Tsichritzis 1998). Such artifacts are not exempt from natural laws or behavioral theories. To the contrary, their creation relies on existing *kernel theories* that are applied, tested, modified, and extended through the experience, creativity, intuition, and problem solving capabilities of the researcher (Markus et al. 2002; Walls et al. 1992).

The importance of design is well recognized in the IS literature (Glass 1999; Winograd 1996, 1998). Benbasat and Zmud (1999, p. 5) argue that the relevance of IS research is directly related to its applicability in design, stating that the implications of empirical IS research should be “implementable, ... synthesize an existing body of research, ... [or] stimulate critical thinking” among IS practitioners. However, designing useful artifacts is complex due to the need for creative advances in domain areas in which existing theory is often insufficient. “As technical knowledge grows, IT is applied to new application areas that were not previously believed to be amenable to IT support” (Markus et al. 2002, p. 180). The resultant IT artifacts extend the boundaries of human problem solving and organizational capabilities by providing intellectual as well as computational tools. Theories regarding their application and impact will follow their development and use.

Here, we argue, is an opportunity for IS research to make significant contributions by engaging the complementary research cycle between design-

science and behavioral-science to address fundamental problems faced in the productive application of information technology. Technology and behavior are not dichotomous in an information system. They are inseparable (Lee 2000). They are similarly inseparable in IS research. Philosophically these arguments draw from the pragmatists (Abouafia 1991) who argue that truth (justified theory) and utility (artifacts that are effective) are two sides of the same coin and that scientific research should be evaluated in light of its practical implications.

The realm of IS research is at the confluence of people, organizations, and technology (Davis and Olson 1985; Lee 1999). IT artifacts are broadly defined as *constructs* (vocabulary and symbols), *models* (abstractions and representations), *methods* (algorithms and practices), and *instantiations* (implemented and prototype systems). These are concrete prescriptions that enable IT researchers and practitioners to understand and address the problems inherent in developing and successfully implementing information systems within organizations (March and Smith 1995; Nunamaker et al. 1991a). As illustrations, Markus et al. (2002) and Walls et al. (1992) present design-science research aimed at developing executive information systems (EISs) and systems to support emerging knowledge processes (EKPs), respectively, within the context of "IS design theories." Such theories prescribe "effective *development practices*" (methods) and "a type of *system solution*" (instantiation) for "a particular class of *user requirements*" (models) (Markus et al. 2002, p. 180). Such prescriptive theories must be evaluated with respect to the utility provided for the class of problems addressed.

An IT artifact, implemented in an organizational context, is often the object of study in IS behavioral-science research. Theories seek to predict or explain phenomena that occur with respect to the artifact's use (intention to use), perceived usefulness, and impact on individuals and organizations (net benefits) depending on system, service, and information quality (DeLone and McLean 1992, 2003; Seddon 1997). Much of this behavioral research has focused on one class of

artifact, the instantiation (system), although other research efforts have also focused on the evaluation of constructs (e.g., Batra et al. 1990; Bodart et al. 2001; Geerts and McCarthy 2002; Kim and March 1995) and methods (e.g., Marakas and Elam 1998; Sinha and Vessey 1999). Relatively little behavioral research has focused on evaluating models, a major focus of research in the management science literature.

Design science, as the other side of the IS research cycle, creates and evaluates IT artifacts intended to solve identified organizational problems. Such artifacts are represented in a structured form that may vary from software, formal logic, and rigorous mathematics to informal natural language descriptions. A mathematical basis for design allows many types of quantitative evaluations of an IT artifact, including optimization proofs, analytical simulation, and quantitative comparisons with alternative designs. The further evaluation of a new artifact in a given organizational context affords the opportunity to apply empirical and qualitative methods. The rich phenomena that emerge from the interaction of people, organizations, and technology may need to be qualitatively assessed to yield an understanding of the phenomena adequate for theory development or problem solving (Klein and Meyers 1999). As field studies enable behavioral-science researchers to understand organizational phenomena in context, the process of constructing and exercising innovative IT artifacts enable design-science researchers to understand the problem addressed by the artifact and the feasibility of their approach to its solution (Nunamaker et al. 1991a).

The primary goal of this paper is to inform the community of IS researchers and practitioners of how to conduct, evaluate, and present design-science research. We do so by describing the boundaries of design science within the IS discipline via a conceptual framework for understanding information systems research and by developing a set of guidelines for conducting and evaluating good design-science research. We focus primarily on technology-based design although we note with interest the current explora-

tion of organizations, policies, and work practices as designed artifacts (Boland 2002). Following Klein and Myers (1999) treatise on the conduct and evaluation of interpretive research in IS, we use the proposed guidelines to assess recent exemplar papers published in the IS literature in order to illustrate how authors, reviewers, and editors can apply them consistently. We conclude with an analysis of the challenges of performing high-quality design-science research and a call for synergistic efforts between behavioral-science and design-science researchers.

A Framework for IS Research ■

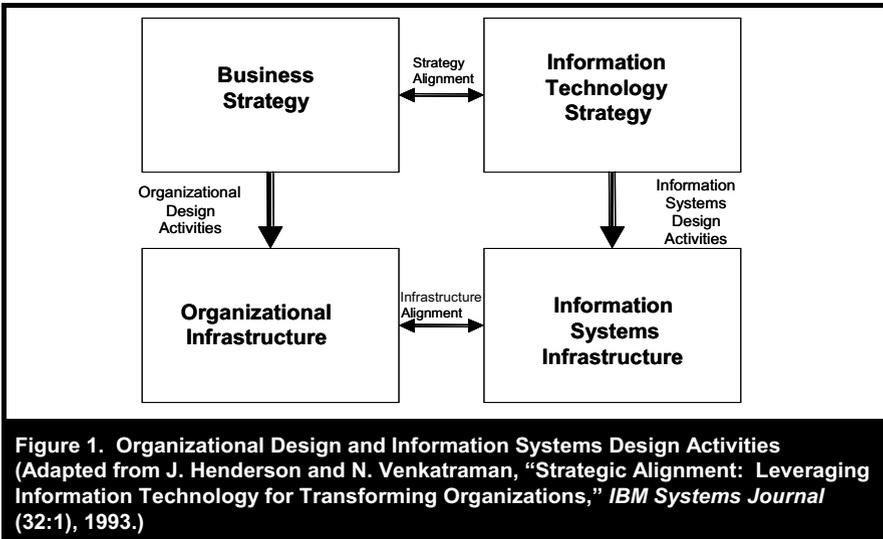
Information systems and the organizations they support are complex, artificial, and purposefully designed. They are composed of people, structures, technologies, and work systems (Alter 2003; Bunge 1985; Simon 1996). Much of the work performed by IS practitioners, and managers in general (Boland 2002), deals with design—the purposeful organization of resources to accomplish a goal. Figure 1 illustrates the essential alignments between business and information technology strategies and between organizational and information systems infrastructures (Henderson and Venkatraman 1993). The effective transition of strategy into infrastructure requires extensive design activity on both sides of the figure—organizational design to create an effective organizational infrastructure and information systems design to create an effective information system infrastructure.

These are interdependent design activities that are central to the IS discipline. Hence, IS research must address the interplay among business strategy, IT strategy, organizational infrastructure, and IS infrastructure. This interplay is becoming more crucial as information technologies are seen as enablers of business strategy and organizational infrastructure (Kalakota and Robinson 2001; Orlikowski and Barley 2001). Available and emerging IT capabilities are a significant factor in determining the strategies that guide an organization. Cutting-edge information systems allow

organizations to engage new forms and new structures—to change the ways they “do business” (Drucker 1988, 1991; Orlikowski 2000). Our subsequent discussion of design science will be limited to the activities of building the IS infrastructure within the business organization. Issues of strategy, alignment, and organizational infrastructure design are outside the scope of this paper.

To achieve a true understanding of and appreciation for design science as an IS research paradigm, an important dichotomy must be faced. Design is both a process (set of activities) and a product (artifact)—a verb and a noun (Walls et al. 1992). It describes the world as acted upon (*processes*) and the world as sensed (*artifacts*). This Platonic view of design supports a problem-solving paradigm that continuously shifts perspective between design processes and designed artifacts for the same complex problem. The design process is a sequence of expert activities that produces an innovative product (i.e., the design artifact). The evaluation of the artifact then provides feedback information and a better understanding of the problem in order to improve both the quality of the product and the design process. This build-and-evaluate loop is typically iterated a number of times before the final design artifact is generated (Markus et al. 2002). During this creative process, the design-science researcher must be cognizant of evolving both the design process and the design artifact as part of the research.

March and Smith (1995) identify two design processes and four design artifacts produced by design-science research in IS. The two processes are *build* and *evaluate*. The artifacts are *constructs*, *models*, *methods*, and *instantiations*. Purposeful artifacts are built to address heretofore unsolved problems. They are evaluated with respect to the utility provided in solving those problems. Constructs provide the language in which problems and solutions are defined and communicated (Schön 1983). Models use constructs to represent a real world situation—the design problem and its solution space (Simon 1996). Models aid problem and solution understanding and frequently represent the connection



between problem and solution components enabling exploration of the effects of design decisions and changes in the real world. Methods define processes. They provide guidance on how to solve problems, that is, how to search the solution space. These can range from formal, mathematical algorithms that explicitly define the search process to informal, textual descriptions of "best practice" approaches, or some combination. Instantiations show that constructs, models, or methods can be implemented in a working system. They demonstrate feasibility, enabling concrete assessment of an artifact's suitability to its intended purpose. They also enable researchers to learn about the real world, how the artifact affects it, and how users appropriate it.

Figure 2 presents our conceptual framework for understanding, executing, and evaluating IS research combining behavioral-science and design-science paradigms. We use this framework to position and compare these paradigms.

The environment defines the problem space (Simon 1996) in which reside the phenomena of interest. For IS research, it is composed of

people, (business) organizations, and their existing or planned technologies (Silver et al. 1995). In it are the goals, tasks, problems, and opportunities that define business needs as they are perceived by people within the organization. Such perceptions are shaped by the roles, capabilities, and characteristics of people within the organization. Business needs are assessed and evaluated within the context of organizational strategies, structure, culture, and existing business processes. They are positioned relative to existing technology infrastructure, applications, communication architectures, and development capabilities. Together these define the business need or "problem" as perceived by the researcher. Framing research activities to address business needs assures research relevance.

Given such an articulated business need, IS research is conducted in two complementary phases. Behavioral science addresses research through the *development* and *justification* of theories that explain or predict phenomena related to the identified business need. Design science addresses research through the *building* and *evaluation* of artifacts designed to meet the iden-

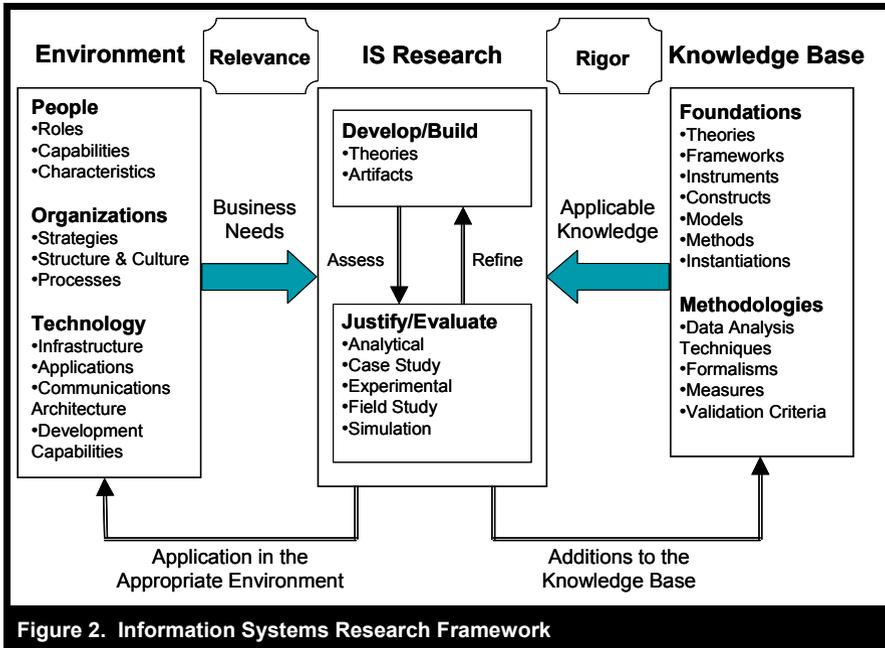


Figure 2. Information Systems Research Framework

tified business need. The goal of behavioral-science research is truth.² The goal of design-science research is utility. As argued above, our position is that truth and utility are inseparable. Truth informs design and utility informs theory. An artifact may have utility because of some as yet undiscovered truth. A theory may yet to be developed to the point where its truth can be incorporated into design. In both cases, research assessment via the justify/evaluate activities can result in the identification of weaknesses in the theory or

artifact and the need to refine and reassess. The refinement and reassessment process is typically described in future research directions.

The knowledge base provides the raw materials from and through which IS research is accomplished. The knowledge base is composed of foundations and methodologies. Prior IS research and results from reference disciplines provide foundational theories, frameworks, instruments, constructs, models, methods, and instantiations used in the develop/build phase of a research study. Methodologies provide guidelines used in the justify/evaluate phase. Rigor is achieved by appropriately applying existing foundations and methodologies. In behavioral science, methodologies are typically rooted in data collection and empirical analysis techniques. In design science, computational and mathematical methods are

²Theories posed in behavioral science are principled explanations of phenomena. We recognize that such theories are approximations and are subject to numerous assumptions and conditions. However, they are evaluated against the norms of truth or explanatory power and are valued only as the claims they make are borne out in reality.

primarily used to evaluate the quality and effectiveness of artifacts; however, empirical techniques may also be employed.

The contributions of behavioral science and design science in IS research are assessed as they are applied to the business need in an appropriate environment and as they add to the content of the knowledge base for further research and practice. A justified theory that is not useful for the environment contributes as little to the IS literature as an artifact that solves a nonexistent problem.

One issue that must be addressed in design-science research is differentiating routine design or system building from design research. The difference is in the nature of the problems and solutions. Routine design is the application of existing knowledge to organizational problems, such as constructing a financial or marketing information system using best practice artifacts (constructs, models, methods, and instantiations) existing in the knowledge base. On the other hand, design-science research addresses important unsolved problems in unique or innovative ways or solved problems in more effective or efficient ways. The key differentiator between routine design and design research is the clear identification of a contribution to the archival knowledge base of foundations and methodologies.

In the early stages of a discipline or with significant changes in the environment, each new artifact created for that discipline or environment is "an experiment" that "poses a question to nature" (Newell and Simon 1976, p 114). Existing knowledge is used where appropriate; however, often the requisite knowledge is nonexistent (Markus et al. 2002). Reliance on creativity and trial-and-error search are characteristic of such research efforts. As design-science research results are codified in the knowledge base, they become best practice. System building is then the routine application of the knowledge base to known problems.

Design activities are endemic in many professions. In particular, the engineering profession

has produced a considerable literature on design (Dym 1994; Pahl and Beitz 1996; Petroski 1996). Within the IS discipline, many design activities have been extensively studied, formalized, and become normal or routine. Design-science research in IS addresses what are considered to be *wicked problems* (Brooks 1987, 1996; Rittel and Webber 1984). That is, those problems characterized by

- unstable requirements and constraints based upon ill-defined environmental contexts
- complex interactions among subcomponents of the problem and its solution
- inherent flexibility to change design processes as well as design artifacts (i.e., malleable processes and artifacts)
- a critical dependence upon human cognitive abilities (e.g., creativity) to produce effective solutions
- a critical dependence upon human social abilities (e.g., teamwork) to produce effective solutions

As a result, we agree with Simon (1996) that a theory of design in information systems, of necessity, is in a constant state of scientific revolution (Kuhn 1996). Technological advances are the result of innovative, creative design science processes. If not capricious, they are at least arbitrary (Brooks 1987) with respect to business needs and existing knowledge. Innovations, such as database management systems, high-level languages, personal computers, software components, intelligent agents, object technology, the Internet, and the World Wide Web, have had dramatic and at times unintended impacts on the way in which information systems are conceived, designed, implemented, and managed. Consequently the guidelines we present below are, of necessity, adaptive and process-oriented.

Guidelines for Design Science in Information Systems Research

As discussed above, design science is inherently a problem solving process. The fundamental principle of design-science research from which our seven guidelines are derived is that knowledge and understanding of a design problem and its solution are acquired in the building and application of an artifact. That is, design-science research requires the creation of an innovative, purposeful artifact (Guideline 1) for a specified problem domain (Guideline 2). Because the artifact is *purposeful*, it must yield utility for the specified problem. Hence, thorough evaluation of the artifact is crucial (Guideline 3). Novelty is similarly crucial since the artifact must be *innovative*, solving a heretofore unsolved problem or solving a known problem in a more effective or efficient manner (Guideline 4). In this way, design-science research is differentiated from the practice of design. The artifact itself must be rigorously defined, formally represented, coherent, and internally consistent (Guideline 5). The process by which it is created, and often the artifact itself, incorporates or enables a search process whereby a problem space is constructed and a mechanism posed or enacted to find an effective solution (Guideline 6). Finally, the results of the design-science research must be communicated effectively (Guideline 7) both to a technical audience (researchers who will extend them and practitioners who will implement them) and to a managerial audience (researchers who will study them in context and practitioners who will decide if they should be implemented within their organizations).

Our purpose for establishing these seven guidelines is to assist researchers, reviewers, editors, and readers to understand the requirements for effective design-science research. Following Klein and Myers (1999), we advise against mandatory or rote use of the guidelines. Researchers, reviewers, and editors must use their creative skills and judgment to determine when, where, and how to apply each of the guidelines in a specific research project. However, we

contend that each of these guidelines should be addressed in some manner for design-science research to be complete. How well the research satisfies the intent of each of the guidelines is then a matter for the reviewers, editors, and readers to determine.

Table 1 summarizes the seven guidelines. Each is discussed in detail below. In the following section, they are applied to specific exemplar research efforts.

Guideline 1: Design as an Artifact

The result of design-science research in IS is, by definition, a purposeful IT artifact created to address an important organizational problem. It must be described effectively, enabling its implementation and application in an appropriate domain.

Orlikowski and Iacono (2001) call the IT artifact the “core subject matter” of the IS field. Although they articulate multiple definitions of the term *IT artifact*, many of which include components of the organization and people involved in the use of a computer-based artifact, they emphasize the importance of “those bundles of cultural properties packaged in some socially recognizable form such as hardware and software” (p. 121), i.e., the IT artifact as an instantiation. Weber (1987) argues that theories of long-lived artifacts (instantiations) and their representations (Weber 2003) are fundamental to the IS discipline. Such theories must explain how artifacts are created and adapted to their changing environments and underlying technologies.

Our definition of IT artifacts is both broader and narrower than those articulated above. It is broader in the sense that we include not only instantiations in our definition of the IT artifact but also the constructs, models, and methods applied in the development and use of information systems. However, it is narrower in the sense that we do not include people or elements of organizations in our definition nor do we explicitly include the process by which such artifacts evolve

Table 1. Design-Science Research Guidelines

Guideline	Description
Guideline 1: Design as an Artifact	Design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation.
Guideline 2: Problem Relevance	The objective of design-science research is to develop technology-based solutions to important and relevant business problems.
Guideline 3: Design Evaluation	The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.
Guideline 4: Research Contributions	Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.
Guideline 5: Research Rigor	Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.
Guideline 6: Design as a Search Process	The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.
Guideline 7: Communication of Research	Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.

over time. We conceive of IT artifacts not as independent of people or the organizational and social contexts in which they are used but as interdependent and coequal with them in meeting business needs. We acknowledge that perceptions and fit with an organization are crucial to the successful development and implementation of an information system. We argue, however, that the capabilities of the constructs, models, methods, and instantiations are equally crucial and that design-science research efforts are necessary for their creation.

Furthermore, artifacts constructed in design-science research are rarely full-grown information systems that are used in practice. Instead, artifacts are innovations that define the ideas, practices, technical capabilities, and products through which the analysis, design, implementation, and use of information systems can be effectively and efficiently accomplished (Denning

1997; Tsichritzis 1998). This definition of the artifact is consistent with the concept of IS design theory as used by Walls et al. (1992) and Markus et al. (2002) where the theory addresses both the process of design and the designed product.

More precisely, constructs provide the vocabulary and symbols used to define problems and solutions. They have a significant impact on the way in which tasks and problems are conceived (Boland 2002; Schön 1983). They enable the construction of models or representations of the problem domain. Representation has a profound impact on design work. The field of mathematics was revolutionized, for example, with the constructs defined by Arabic numbers, zero, and place notation. The search for an effective problem representation is crucial to finding an effective design solution (Weber 2003). Simon (1996, p. 132) states, "solving a problem simply means representing it so as to make the solution transparent."

The entity-relationship model (Chen 1976), for example, is a set of constructs for representing the semantics of data. It has had a profound impact on the way in which systems analysis and database design are executed and the way in which information systems are represented and developed. Furthermore, these constructs have been used to build models of specific business situations that have been generalized into patterns for application in similar domains (Purao et al. 2003). Methods for building such models have also been the subject of considerable research (Halpin 2001; McCarthy 1982; Parsons and Wand 2000; Storey et al. 1997).

Artifact instantiation demonstrates feasibility both of the design process and of the designed product. Design-science research in IT often addresses problems related to some aspect of the *design* of an information system. Hence, the instantiations produced may be in the form of intellectual or software tools aimed at improving the process of information system development. Constructing a system instantiation that automates a process demonstrates that the process can, in fact, be automated. It provides “proof by construction” (Nunamaker 1991a). The critical nature of design-science research in IS lies in the identification of as yet undeveloped capabilities needed to expand IS into new realms “not previously believed amenable to IT support” (Markus et al. 2002, p. 180). Such a result is significant IS research only if there is a serious question about the ability to construct such an artifact, there is uncertainty about its ability to perform appropriately, and the automated task is important to the IS community. TOP Modeler (Markus et al. 2002), for example, is a tool that instantiates methods for the development of information systems that support “emergent knowledge processes.” Construction of such a prototype artifact in a research setting or in a single organizational setting is only a first step toward its deployment, but we argue that it is a necessary one. As an exemplar of design-science research (see below), this research resulted in a commercial product that “has been used in over two dozen ‘real use’ situations” (p. 187).

To illustrate further, prior to the construction of the first expert system (instantiation), it was not clear if such a system *could* be constructed. It was not clear how to describe or represent it, or how well it would perform. Once feasibility was demonstrated by constructing an expert system in a selected domain, constructs and models were developed and subsequent research in expert systems focused on demonstrating significant improvements in the product or process (methods) of construction (Tam 1990; Trice and Davis 1993). Similar examples exist in requirements determination (Bell 1993; Bhargava et al. 1998), individual and group decision support systems (Aiken et al. 1991; Basu and Blanning 1994), database design and integration (Dey et al. 1998; Dey et al. 1999; Storey et al. 1997), and workflow analysis (Basu and Blanning 2000), to name a few important areas of IS design-science research.

Guideline 2: Problem Relevance

The objective of research in information systems is to acquire knowledge and understanding that enable the development and implementation of technology-based solutions to heretofore unsolved and important business problems. Behavioral science approaches this goal through the development and justification of theories explaining or predicting phenomena that occur. Design science approaches this goal through the construction of innovative artifacts aimed at changing the phenomena that occur. Each must inform and challenge the other. For example, the technology acceptance model provides a theory that explains and predicts the acceptance of information technologies within organizations (Venkatesh 2000). This theory challenges design-science researchers to create artifacts that enable organizations to overcome the acceptance problems predicted. We argue that a combination of technology-based artifacts (e.g., system conceptualizations and representations, practices, technical capabilities, interfaces, etc.), organization-based artifacts (e.g., structures, compensation, reporting relationships, social systems, etc.), and people-based artifacts (e.g., training, consensus building, etc.) are necessary to address such issues.

Formally, a problem can be defined as the differences between a goal state and the current state of a system. Problem solving can be defined as a search process (see Guideline 6) using actions to reduce or eliminate the differences (Simon 1996). These definitions imply an environment that imposes goal criteria as well as constraints upon a system. Business organizations are goal-oriented entities existing in an economic and social setting. Economic theory often portrays the goals of business organizations as being related to profit (utility) maximization. Hence, business problems and opportunities often relate to increasing revenue or decreasing cost through the design of effective business processes. The design of organizational and inter-organizational information systems plays a major role in enabling effective business processes to achieve these goals.

The relevance of any design-science research effort is with respect to a constituent community. For IS researchers, that constituent community is the practitioners who plan, manage, design, implement, operate, and evaluate information systems and those who plan, manage, design, implement, operate, and evaluate the technologies that enable their development and implementation. To be relevant to this community, research must address the problems faced and the opportunities afforded by the interaction of people, organizations, and information technology. Organizations spend billions of dollars annually on IT, only too often to conclude that those dollars were wasted (Keil 1995; Keil et al. 1998; Keil and Robey 1999). This community would welcome effective artifacts that enable such problems to be addressed—constructs by which to think about them, models by which to represent and explore them, methods by which to analyze or optimize them, and instantiations that demonstrate how to affect them.

Guideline 3: Design Evaluation

The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods. Evaluation is a

crucial component of the research process. The business environment establishes the requirements upon which the evaluation of the artifact is based. This environment includes the technical infrastructure which itself is incrementally built by the implementation of new IT artifacts. Thus, evaluation includes the integration of the artifact within the technical infrastructure of the business environment.

As in the justification of a behavioral science theory, evaluation of a designed IT artifact requires the definition of appropriate metrics and possibly the gathering and analysis of appropriate data. IT artifacts can be evaluated in terms of functionality, completeness, consistency, accuracy, performance, reliability, usability, fit with the organization, and other relevant quality attributes. When analytical metrics are appropriate, designed artifacts may be mathematically evaluated. As two examples, distributed database design algorithms can be evaluated using expected operating cost or average response time for a given characterization of information processing requirements (Johansson et al. 2003) and search algorithms can be evaluated using information retrieval metrics such as precision and recall (Salton 1988).

Because design is inherently an iterative and incremental activity, the evaluation phase provides essential feedback to the construction phase as to the quality of the design process and the design product under development. A design artifact is complete and effective when it satisfies the requirements and constraints of the problem it was meant to solve. Design-science research efforts may begin with simplified conceptualizations and representations of problems. As available technology or organizational environments change, assumptions made in prior research may become invalid. Johansson (2000), for example, demonstrated that network latency is a major component in the response-time performance of distributed databases. Prior research in distributed database design ignored latency because it assumed a low-bandwidth network where latency is negligible. In a high-bandwidth network, however, latency can account for over 90

1. Observational	Case Study: Study artifact in depth in business environment
	Field Study: Monitor use of artifact in multiple projects
2. Analytical	Static Analysis: Examine structure of artifact for static qualities (e.g., complexity)
	Architecture Analysis: Study fit of artifact into technical IS architecture
	Optimization: Demonstrate inherent optimal properties of artifact or provide optimality bounds on artifact behavior
	Dynamic Analysis: Study artifact in use for dynamic qualities (e.g., performance)
3. Experimental	Controlled Experiment: Study artifact in controlled environment for qualities (e.g., usability)
	Simulation – Execute artifact with artificial data
4. Testing	Functional (Black Box) Testing: Execute artifact interfaces to discover failures and identify defects
	Structural (White Box) Testing: Perform coverage testing of some metric (e.g., execution paths) in the artifact implementation
5. Descriptive	Informed Argument: Use information from the knowledge base (e.g., relevant research) to build a convincing argument for the artifact's utility
	Scenarios: Construct detailed scenarios around the artifact to demonstrate its utility

percent of the response time. Johansson et al. (2003) extended prior distributed database design research by developing a model that includes network latency and the effects of parallel processing on response time.

The evaluation of designed artifacts typically uses methodologies available in the knowledge base. These are summarized in Table 2. The selection of evaluation methods must be matched appropriately with the designed artifact and the selected evaluation metrics. For example, descriptive methods of evaluation should only be used for especially innovative artifacts for which other forms of evaluation may not be feasible. The goodness and efficacy of an artifact can be rigorously demonstrated via well-selected evaluation methods (Basili 1996; Kleindorfer et al. 1998; Zelkowitz and Wallace 1998).

Design, in all of its realizations (e.g., architecture, landscaping, art, music), has style. Given the problem and solution requirements, sufficient degrees of freedom remain to express a variety of forms and functions in the artifact that are aesthetically pleasing to both the designer and the user. Good designers bring an element of style to their work (Norman 1988). Thus, we posit that design evaluation should include an assessment of the artifact's style.

The measurement of style lies in the realm of human perception and taste. In other words, we know good style when we see it. While difficult to define, style in IS design is widely recognized and appreciated (Kernighan and Plauger 1978; Winograd 1996). Gelerter (1998) terms the essence of style in IS design *machine beauty*. He describes it as a marriage between simplicity and

power that drives innovation in science and technology. Simon (1996) also notes the importance of style in the design process. The ability to creatively vary the design process, within the limits of satisfactory constraints, challenges and adds value to designers who participate in the process.

Guideline 4: Research Contributions

Effective design-science research must provide clear contributions in the areas of the design artifact, design construction knowledge (i.e., foundations), and/or design evaluation knowledge (i.e., methodologies). The ultimate assessment for any research is, "What are the new and interesting contributions?" Design-science research holds the potential for three types of research contributions based on the novelty, generality, and significance of the designed artifact. One or more of these contributions must be found in a given research project.

1. *The Design Artifact.* Most often, the contribution of design-science research is the artifact itself. The artifact must enable the solution of heretofore unsolved problems. It may extend the knowledge base (see below) or apply existing knowledge in new and innovative ways. As shown in Figure 2 by the left-facing arrow at the bottom of the figure from IS Research to the Environment, exercising the artifact in the environment produces significant value to the constituent IS community. System development methodologies, design tools, and prototype systems (e.g., GDSS, expert systems) are examples of such artifacts.
2. *Foundations.* The creative development of novel, appropriately evaluated constructs, models, methods, or instantiations that extend and improve the existing foundations in the design-science knowledge base are also important contributions. The right-facing arrow at the bottom of the figure from IS Research to the Knowledge Base in Figure 2 indicates these contributions. Modeling

formalisms, ontologies (Wand and Weber 1993, 1995; Weber 1997), problem and solution representations, design algorithms (Storey et al. 1997), and innovative information systems (Aiken 1991; Markus et al. 2002; Walls et al. 1992) are examples of such artifacts.

3. *Methodologies.* Finally, the creative development and use of evaluation methods (e.g., experimental, analytical, observational, testing, and descriptive) and new evaluation metrics provide design-science research contributions. Measures and evaluation metrics in particular are crucial components of design-science research. The right-facing arrow at the bottom of the figure from IS Research to the Knowledge Base in Figure 2 also indicates these contributions. TAM, for example, presents a framework for predicting and explaining why a particular information system will or will not be accepted in a given organizational setting (Venkatesh 2000). Although TAM is posed as a behavioral theory, it also provides metrics by which a designed information system or implementation process can be evaluated. Its implications for design itself are as yet unexplored.

Criteria for assessing contribution focus on *representational fidelity* and *implementability*. Artifacts must accurately represent the business and technology environments used in the research, information systems themselves being models of the business. These artifacts must be "implementable," hence the importance of instantiating design science artifacts. Beyond these, however, the research must demonstrate a clear contribution to the business environment, solving an important, previously unsolved problem.

Guideline 5: Research Rigor

Rigor addresses the way in which research is conducted. Design-science research requires the application of rigorous methods in both the construction and evaluation of the designed artifact. In behavioral-science research, rigor is

often assessed by adherence to appropriate data collection and analysis techniques. Overemphasis on rigor in behavioral IS research has often resulted in a corresponding lowering of relevance (Lee 1999).

Design-science research often relies on mathematical formalism to describe the specified and constructed artifact. However, the environments in which IT artifacts must perform and the artifacts themselves may defy excessive formalism. Or, in an attempt to be mathematically rigorous, important parts of the problem may be abstracted or “assumed away.” In particular, with respect to the construction activity, rigor must be assessed with respect to the applicability and generalizability of the artifact. Again, an overemphasis on rigor can lessen relevance. We argue, along with behavioral IS researchers (Applegate 1999), that it is possible and necessary for all IS research paradigms to be both rigorous and relevant.

In both design-science and behavioral-science research, rigor is derived from the effective use of the knowledge base—theoretical foundations and research methodologies. Success is predicated on the researcher’s skilled selection of appropriate techniques to develop or construct a theory or artifact and the selection of appropriate means to justify the theory or evaluate the artifact.

Claims about artifacts are typically dependent upon performance metrics. Even formal mathematical proofs rely on evaluation criteria against which the performance of an artifact can be measured. Design-science researchers must constantly assess the appropriateness of their metrics and the construction of effective metrics is an important part of design-science research.

Furthermore, designed artifacts are often components of a human-machine problem-solving system. For such artifacts, knowledge of behavioral theories and empirical work are necessary to construct and evaluate such artifacts. Constructs, models, methods, and instantiations must be exercised within appropriate environments. Appropriate subject groups must be obtained for such studies. Issues that are addressed include

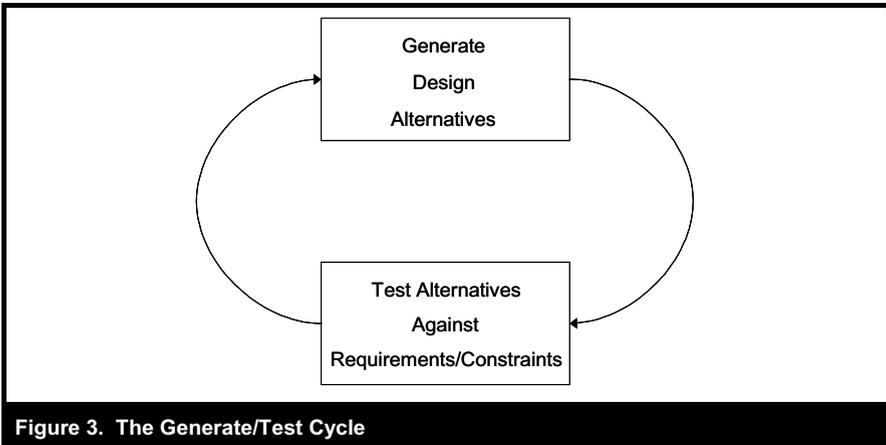
comparability, subject selection, training, time, and tasks. Methods for this type of evaluation are not unlike those for justifying or testing behavioral theories. However, the principal aim is to determine how well an artifact works, not to theorize about or prove anything about why the artifact works. This is where design-science and behavioral-science researchers must complement one another. Because design-science artifacts are often the “machine” part of the human-machine system constituting an information system, it is imperative to understand why an artifact works or does not work to enable new artifacts to be constructed that exploit the former and avoid the latter.

Guideline 6: Design as a Search Process

Design science is inherently iterative. The search for the best, or optimal, design is often intractable for realistic information systems problems. Heuristic search strategies produce feasible, good designs that can be implemented in the business environment. Simon (1996) describes the nature of the design process as a Generate/Test Cycle (Figure 3).

Design is essentially a search process to discover an effective solution to a problem. Problem solving can be viewed as utilizing available means to reach desired ends while satisfying laws existing in the environment (Simon 1996). Abstraction and representation of appropriate means, ends, and laws are crucial components of design-science research. These factors are problem and environment dependent and invariably involve creativity and innovation. Means are the set of actions and resources available to construct a solution. Ends represent goals and constraints on the solution. Laws are uncontrollable forces in the environment. Effective design requires knowledge of both the application domain (e.g., requirements and constraints) and the solution domain (e.g., technical and organizational).

Design-science research often simplifies a problem by explicitly representing only a subset of the



relevant means, ends, and laws or by decomposing a problem into simpler subproblems. Such simplifications and decompositions may not be realistic enough to have a significant impact on practice but may represent a starting point. Progress is made iteratively as the scope of the design problem is expanded. As means, ends, and laws are refined and made more realistic, the design artifact becomes more relevant and valuable. The means, ends, and laws for IS design problems can often be represented using the tools of mathematics and operations research. Means are represented by decision variables whose values constitute an implementable design solution. Ends are represented using a utility function and constraints that can be expressed in terms of decision variables and constants. Laws are represented by the values of constants used in the utility function and constraints.

The set of possible design solutions for any problem is specified as all possible means that satisfy all end conditions consistent with identified laws. When these can be formulated appropriately and posed mathematically, standard operations research techniques can be used to determine an optimal solution for the specified end conditions. Given the wicked nature of many information system design problems, however, it

may not be possible to determine, let alone explicitly describe, the relevant means, ends, or laws (Vessey and Glass 1998). Even when it is possible to do so, the sheer size and complexity of the solution space will often render the problem computationally infeasible. For example, to build a “reliable, secure, and responsive information systems infrastructure,” one of the key issues faced by IS managers (Brancheau et al. 1996), a designer would need to represent all possible infrastructures (means), determine their utility and constraints (ends), and specify all cost and benefit constants (laws). Clearly such an approach is infeasible. However, this does not mean that design-science research is inappropriate for such a problem.

In such situations, the search is for satisfactory solutions, i.e., *satisficing* (Simon 1996), without explicitly specifying all possible solutions. The design task involves the creation, utilization, and assessment of heuristic search strategies. That is, constructing an artifact that “works” well for the specified class of problems. Although its construction is based on prior theory and existing design knowledge, it may or may not be entirely clear why it works or the extent of its generalizability; it simply qualifies as “credentialed knowledge” (Meehl 1986, p. 311). While it is important

to understand why an artifact works, the critical nature of design in IS makes it important to first establish that it *does* work and to characterize the environments in which it works, even if we cannot completely explain *why* it works. This enables IS practitioners to take advantage of the artifact to improve practice and provides a context for additional research aimed at more fully explicating the resultant phenomena. Markus et al. (2002), for example, describe their search process in terms of iteratively identifying deficiencies in constructed prototype software systems and creatively developing solutions to address them.

The use of heuristics to find “good” design solutions opens the question of how goodness is measured. Different problem representations may provide varying techniques for measuring how good a solution is. One approach is to prove or demonstrate that a heuristic design solution is always within close proximity of an optimal solution. Another is to compare produced solutions with those constructed by expert human designers for the same problem situation.

Guideline 7: Communication of Research

Design-science research must be presented both to technology-oriented as well as management-oriented audiences. Technology-oriented audiences need sufficient detail to enable the described artifact to be constructed (implemented) and used within an appropriate organizational context. This enables practitioners to take advantage of the benefits offered by the artifact and it enables researchers to build a cumulative knowledge base for further extension and evaluation. It is also important for such audiences to understand the processes by which the artifact was constructed and evaluated. This establishes repeatability of the research project and builds the knowledge base for further research extensions by design-science researchers in IS.

Management-oriented audiences need sufficient detail to determine if the organizational resources

should be committed to constructing (or purchasing) and using the artifact within their specific organizational context. Zmud (1997) suggests that presentation of design-science research for a managerial audience requires an emphasis not on the inherent nature of the artifact itself, but on the knowledge required to effectively apply the artifact “within specific contexts for individual or organizational gain” (p. ix). That is, the emphasis must be on the importance of the problem and the novelty and effectiveness of the solution approach realized in the artifact. While we agree with this statement, we note that it may be necessary to describe the artifact in some detail to enable managers to appreciate its nature and understand its application. Presenting that detail in concise, well-organized appendices, as advised by Zmud, is an appropriate communication mechanism for such an audience.

Application of the Design Science Research Guidelines

To illustrate the application of the design-science guidelines to IS research, we have selected three exemplar articles for analysis from three different IS journals, one from *Decision Support Systems*, one from *Information Systems Research*, and one from *MIS Quarterly*. Each has strengths and weaknesses when viewed through the lens of the above guidelines. Our goal is not to perform a critical evaluation of the quality of the research contributions, but rather to illuminate the design-science guidelines. The articles are

- Gavish and Gerdes (1998), which develops techniques for implementing anonymity in Group Decision Support Systems (GDSS) environments
- Aalst and Kumar (2003), which proposes a design for an *eXchangeable Routing Language* (XRL) to support electronic commerce workflows among trading partners

- Markus, Majchrzak, and Gasser (2002), which proposes a design theory for the development of information systems built to support emergent knowledge processes

The fundamental questions for design-science research are, "What utility does the new artifact provide?" and "What demonstrates that utility?" Evidence must be presented to address these two questions. That is the essence of design science. Contribution arises from utility. If existing artifacts are adequate, then design-science research that creates a new artifact is unnecessary (it is irrelevant). If the new artifact does not map adequately to the real world (rigor), it cannot provide utility. If the artifact does not solve the problem (search, implementability), it has no utility. If utility is not demonstrated (evaluation), then there is no basis upon which to accept the claims that it provides any contribution (contribution). Furthermore, if the problem, the artifact, and its utility are not presented in a manner such that the implications for research and practice are clear, then publication in the IS literature is not appropriate (communication).

The Design and Implementation of Anonymity in GDSS: Gavish and Gerdes

The study of group decision support systems (GDSS) has been and remains one of the most visible and successful research streams in the IS field. The use of information technology to effectively support meetings of groups of different sizes over time and space is a real problem that challenges all business organizations. Recent GDSS literature surveys demonstrate the large numbers of GDSS research papers published in the IS field and, more importantly, the wide variety of research paradigms applied to GDSS research (e.g., Dennis and Wixom 2001; Fjermestad and Hiltz 1998; Nunamaker et al. 1996). However, only a small number of GDSS papers can be considered to make true design-science research contributions. Most assume the introduction of a new information technology or process in the

GDSS environment and then study the individual, group, or organizational implications using a behavioral-science research paradigm. Several such GDSS papers have appeared in *MIS Quarterly* (e.g., Dickson et al. 1993; Gallupe et al. 1988; Jarvenpaa et al. 1988; Sengupta and Te'eni 1993).

The central role of design science in GDSS is clearly recognized in the early foundation papers of the field. The University of Arizona Electronic Meeting System group, for example, states the need for both *developmental* and *empirical* research agendas (Dennis et al. 1988; Nunamaker et al. 1991b). Developmental, or design-science, research is called for in the areas of process structures and support and task structures and support. Process structure and support technologies and methods are generic to all GDSS environments and tasks. Technologies and methods for distributed communications, group memory, decision-making methods, and anonymity are a few of the critical design issues for GDSS process support needed in any task domain. Task structure and support are specific to the problem domain under consideration by the group (e.g., medical decision making, software development). Task support includes the design of new technologies and methods for managing and analyzing task-related information and using that information to make specific, task-related decisions.

The issue of anonymity has been studied extensively in GDSS environments. Behavioral research studies have shown both positive and negative impacts on group interactions. On the positive side, GDSS participants can express their views freely without fear of embarrassment or reprisal. However, anonymity can encourage free-riding and antisocial behaviors. While the pros and cons of anonymity in GDSS are much researched, there has been a noticeable lack of research on the design of techniques for implementing anonymity in GDSS environments. Gavish and Gerdes (1998) address this issue by designing five basic mechanisms to provide GDSS procedural anonymity.

Problem Relevance

The amount of interest and research on anonymity issues in GDSS testifies to its relevance. Field studies and surveys clearly indicate that participants rank anonymity as a highly desired attribute in the GDSS system. Many individuals state that they would refuse to participate in or trust the results of a GDSS meeting without a satisfactory level of assured anonymity (Fjermestad and Hiltz 1998).

Research Rigor

Gavish and Gerdes base their GDSS anonymity designs on past research in the fields of cryptography and secure network communication protocols (e.g., Chaum 1981; Schneier 1996). These research areas have a long history of formal, rigorous results that have been applied to the design of many practical security and privacy mechanisms. Appendix A of the exemplar paper provides a set of formal proofs that the claims made by the authors for the anonymity designs are correct and draw their validity from the knowledge base of this past research.

Design as a Search Process

The authors motivate their design science research by identifying three basic types of anonymity in a GDSS system: *environmental*, *content*, and *procedural*. After a definition and brief discussion of each type, they focus on the design of mechanisms for procedural anonymity; the ability of the GDSS system to hide the source of any message. This is a very difficult requirement because standard network protocols typically attach source information in headers to support reliable transmission protocols. Thus, GDSS systems must modify standard communication protocols and include additional transmission procedures to ensure required levels of anonymity.

The design-science process employed by the authors is to state the desired procedural anonymity attributes of the GDSS system and then to

design mechanisms to satisfy the system requirements for anonymity. Proposed designs are presented and anonymity claims are proved to be correct. A thorough discussion of the costs and benefits of the proposed anonymity mechanisms is provided in Section 4 of the paper.

Design as an Artifact

The authors design a GDSS system architecture that provides a rigorous level of procedural anonymity. Five mechanisms are employed to ensure participant anonymity:

- All messages are encrypted with a unique session key
- The sender's header information is removed from all messages
- All messages are re-encrypted upon retransmission from any GDSS server
- Transmission order of messages is randomized
- Artificial messages are introduced to thwart traffic analysis

The procedures and communication protocols that implement these mechanisms in a GDSS system are the artifacts of this research.

Design Evaluation

The evaluation consists of two reported activities. First, in Appendix A, each mechanism is proved to correctly provide the claimed anonymity benefits. Formal proof methods are used to validate the effectiveness of the designed mechanisms. Second, Section 4 presents a thorough cost-benefit analysis. It is shown that the operational costs of supporting the proposed anonymity mechanisms can be quite significant. In addition, the communication protocols to implement the mechanisms add considerable complexity to the system. Thus, the authors recommend that a

cost-benefit justification be performed before determining the level of anonymity to implement for a GDSS meeting.

The authors do not claim to have implemented the proposed anonymity mechanisms in a prototype or actual GDSS system. Thus, an instantiation of the designed artifact remains to be evaluated in an operational GDSS environment.

Research Contributions

The design-science contributions of this research are the proposed anonymity mechanisms as the design artifacts and the evaluation results in the form of formal proofs and cost-benefit analyses. These contributions advance our understanding of how best to provide participant anonymity in GDSS meetings.

Research Communication

Although the presentation of this research is aimed at an audience familiar with network system concepts such as encryption and communication protocols, the paper also contains important, useful information for a managerial audience. Managers should have a good understanding of the implications of anonymity in GDSS meetings. This understanding must include an appreciation of the costs of providing desired levels of participant anonymity. While the authors provide a thorough discussion of cost-benefit tradeoffs toward the end of the paper, the paper would be more accessible to a managerial audience if it included a stronger motivation up front on the important implications of anonymity in GDSS system development and operations.

A Workflow Language for Inter-organizational Processes: Aalst and Kumar

Workflow models are an effective means for describing, analyzing, implementing, and managing

business processes. Workflow management systems are becoming integral components of many commercial enterprise-wide information systems (Leymann and Roller 2000). Standards for workflow semantics and syntax (i.e., workflow languages) and workflow architectures are promulgated by the Workflow Management Coalition (WfMC 2000). While workflow models have been used for many years to manage intra-organizational business processes, there is now a great demand for effective tools to model inter-organization processes across heterogeneous and distributed environments, such as those found in electronic commerce and complex supply chains (Kumar and Zhao 2002).

Aalst and Kumar (2003) investigate the problem of exchanging business process information across multiple organizations in an automated manner. They design an eXchangable Routing Language (XRL) to capture workflow models that are then embedded in eXtensible Markup Language (XML) for electronic transmission to all participants in an interorganizational business process. The design of XRL is based upon Petri nets, which provide a formal basis for analyzing the correctness and performance of the workflows, as well as supporting the extensibility of the language. The authors develop a workflow management architecture and a prototype implementation to evaluate XRL in a proof of concept.

Problem Relevance

Interorganizational electronic commerce is growing rapidly and is projected to soon exceed one trillion dollars annually (eMarketer 2002). A multitude of electronic commerce solutions are being proposed (e.g., ebXML, UDDI, RosettaNet) to enable businesses to execute transactions in standardized, open environments. While XML has been widely accepted as a protocol for exchanging business data, there is still no clear standard for exchanging business process information (e.g., workflow models). This is the very relevant problem addressed by this research.

Research Rigor

Research on workflow modeling has long been based on rigorous mathematical techniques such as Markov chains, queueing networks, and Petri nets (Aalst and Hee 2002). In this paper, Petri nets provide the underlying semantics for XRL. These formal semantics allow for powerful analysis techniques (e.g., correctness, performance) to be applied to the designed workflow models. Such formalisms also enable the development of automated tools to manipulate and analyze complex workflow designs. Each language construct in XRL has an equivalent Petri-net representation presented in the paper. The language is extensible in that adding a new construct simply requires defining its Petri-net representation and adding its syntax to the XRL. Thus, this research draws from a clearly defined and tested base of modeling literature and knowledge.

Design as a Search Process

XRL is designed in the paper by performing a thorough analysis of business process requirements and identifying features provided by leading commercial workflow management systems. Using the terminology from the paper, workflows traverse routes through available tasks (i.e., business services) in the electronic business environment. The basic routing constructs of XRL define the specific control flow of the business process. The authors build 13 basic constructs into XRL: Task, Sequence, Any_sequence, Choice, Condition, Parallel_sync, Parallel_no_sync, Parallel_part_sync, Wait_all, Wait_any, While_do, Stop, and Terminate. They show the Petri-net representation of each construct. Thus, the fundamental control flow structures of sequence, decision, iteration, and concurrency are supported in XRL.

The authors demonstrate the capabilities of XRL in several examples. However, they are careful not to claim that XRL is *complete* in the formal sense that all possible business processes can be modeled in XRL. The search for a complete set of XRL constructs is left for future research.

Design as an Artifact

There are two clearly identifiable artifacts produced in this research. First, the workflow language XRL is designed. XRL is based on Petri-net formalisms and described in XML syntax. Interorganizational business processes are specified via XRL for execution in a distributed, heterogeneous environment.

The second research artifact is the XRL/flower workflow management architecture in which XRL-described processes are executed. The XRL routing scheme is parsed by an XML parser and stored as an XML data structure. This structure is read into a Petri-net engine which determines the next step of the business process and informs the next task provider via an e-mail message. Results of each task are sent back to the engine which then executes the next step in the process until completion. The paper presents a prototype implementation of the XRL/flower architecture as a proof of concept (Aalst and Kumar 2003).

Another artifact of this research is a workflow verification tool named *Wolfan* that verifies the *soundness* of business process workflows. Soundness of a workflow requires that the workflow terminates, no Petri-net tokens are left behind upon termination, and there are no dead tasks in the workflow. This verification tool is described more completely in a different paper (Aalst 1999).

Design Evaluation

The authors evaluate the XRL and XRL/flower designs in several important ways:

- XRL is compared and contrasted with languages in existing commercial workflow systems and research prototypes. The majority of these languages are proprietary and difficult to adapt to *ad hoc* business process design.
- The fit of XRL with proposed standards is studied. In particular, the *Interoperability Wf-*

XML Binding standard (WfMC 2000) does not at this time include the specification of control flow and, thus, is not suitable for inter-organizational workflows. Electronic commerce standards (e.g., RosettaNet) provide some level of control flow specification for predefined business activities, but do not readily allow the *ad hoc* specification of business processes.

- A research prototype of XRL/flower has been implemented and several of the user interface screens are presented. The screens demonstrate a mail-order routing schema case study.
- The Petri-net foundation of XRL allows the authors to claim the XRL workflows can be verified for correctness and performance. XRL is extensible since new constructs can be added to the language based on their translation to underlying Petri-net representations. However, as discussed above, the authors do not make a formal claim for the representational completeness of XRL.

Research Contributions

The clear contributions of this research are the design artifacts—XRL (a workflow language), XRL/flower (a workflow architecture and its implemented prototype system), and Wolfan (a Petri-net verification engine). Another interesting contribution is the extension of XML in its ability to describe and transmit routing schemas (e.g., control flow information) to support interorganizational electronic commerce.

Research Communication

This paper provides clear information to both technical and managerial audiences. The presentation, while primarily technical with XML coding and Petri-net diagrams throughout, motivates a managerial audience with a strong introduction on risks and benefits of applying interorganizational workflows to electronic commerce applications.

Information Systems Design for Emergent Knowledge Processes: Markus, Majchrzak, and Gasser

Despite decades of research and development efforts, effective methods for developing information systems that meet the information requirements of upper management remain elusive. Early approaches used a “waterfall” approach where requirements were defined and validated prior to initiating design efforts which, in turn, were completed prior to implementation (Royce 1998). Prototyping approaches emerged next, followed by numerous proposals including CASE tool-based approaches, rapid application development, and extreme programming (Kruchten 2000). Walls et al. (1992) propose a framework for a prescriptive information system design theory aimed at enabling designers to construct “more effective information systems” (p. 36). They apply this framework to the design of vigilant executive information systems. The framework establishes a class of user requirements (model of design problems) that are most effectively addressed using a particular type of system solution (instantiation) designed using a prescribed set of development practices (methods). Markus et al. (2002) extend this framework to the development of information systems to support emergent knowledge processes (EKPs)—processes in which structure is “neither possible nor desirable” (p. 182) and where processes are characterized by “*highly unpredictable user types and work contexts*” (p. 183).

Problem Relevance

The relevance and importance of the problem are well demonstrated. Markus et al. describe a class of management activities that they term emergent knowledge processes (EKPs). These include “basic research, new product development, strategic business planning, and organization design” (p. 179). They are characterized by “process emergence, unpredictable user types and use contexts, and distributed expert knowledge” (p. 186). They are crucial to many manufacturing organizations, particularly those in high-tech

industries. Such organizations recognize the need to integrate organizational design and information system design with manufacturing operations. They recognize the potential for significant performance improvements offered by such integration. Yet few have realized that potential. Markus et al. argue that this is due to a lack of an adequate design theory and lack of scientifically based tools, noting that existing information system development methodologies focus on structured or semi-structured decision processes and are inadequate for the development of systems to support EKPs. TOP Modeler, the artifact created in this research effort, squarely addresses this problem. Not surprisingly, its development attracted the attention and active participation of several large, high-tech manufacturing organizations including "Hewlett-Packard, General Motors, Digital Equipment Corporation, and Texas Instruments" (p. 186).

Research Rigor

The presented work has theoretical foundations in both IS design theory and organizational design theory. It uses the basic notions of IS design theory presented in Walls et al. (1992) and poses a prescription for designing information systems to support EKPs. Prior research in developing decision support systems, executive information systems, and expert systems serves as a foundation for this work and deficiencies of these approaches for the examined problem type serve as motivation. The knowledge-base constructed within TOP Modeler was formed from a synthesis of socio-technical systems theory and the empirical literature on organizational design knowledge. It was evaluated theoretically using standard metrics from the expert systems literature and empirically using data gathered from numerous electronics manufacturing companies in the United States. Development of TOP Modeler used an "action research paradigm" starting with a "kernel theory" based on prior development methods and theoretical results and iteratively posing and testing artifacts (prototypes) to assess progress toward the desired result. Finally, the artifact was commercialized and "used

in over two dozen 'real use' situations." (p. 187). In summary, this work effectively used theoretical foundations from IS and organizational theory, applied appropriate research methods in developing the artifact, defined and applied appropriate performance measures, and tested the artifact within an appropriate context.

Design as a Search Process

As discussed above, implementation and iteration are central to this research. The authors study prototypes that instantiate posed or newly learned design prescriptions. Their use and impacts were observed, problems identified, solutions posed and implemented, and the cycle was then repeated. These interventions occurred over a period of 18 months within the aforementioned companies as they dealt with organizational design tasks. As a result, not only was the TOP Modeler developed and deployed but prescriptions (methods) in the form of six principles for developing systems to support EKPs were also devised. The extensive experience, creativity, intuition, and problem solving capabilities of the researchers were involved in assessing problems and interpreting the results of deploying various TOP modeler iterations and in constructing improvements to address shortcomings identified.

Design as an Artifact

The TOP Modeler is an implemented software system (instantiation). It is composed of an object-oriented user interface, an object-oriented query generator, and an analysis module built on top of a relational meta-knowledge base that enables access to "pluggable" knowledge bases representing different domains. It also includes tools to support the design and construction of these knowledge bases. The TOP Modeler supports a development process incorporating the six principles for developing systems to support EKPs. As mentioned above, TOP Modeler was commercialized and used in a number of different organizational redesign situations.

Design Evaluation

Evaluation is in the context of organizational design in manufacturing organizations, and is based on observation during the development and deployment of a single artifact, TOP Modeler. No formal evaluation was attempted in the sense of comparison with other artifacts. This is not surprising, nor is it a criticism of this work. There simply are no existing artifacts that address the same problem. However, given that methodologies for developing information systems to support semi-structured management activities are the closest available artifacts, it is appropriate to use them as a comparative measure. In effect, this was accomplished by using principles from these methodologies to inform the initial design of TOP Modeler. The identification of deficiencies in the resultant artifact provides evidence that these artifacts are ill-suited to the task at hand.

Iterative development and deployment within the context of organizational design in manufacturing organizations provide opportunities to observe improvement but do not enable formal evaluation—at each iteration, changes are induced in the organization that cannot be controlled. As mentioned above, the authors have taken a creative and innovative approach that, of necessity, trades off rigor for relevancy. In the initial stages of a discipline, this approach is extremely effective. TOP Modeler demonstrates the feasibility of developing an artifact to support organizational design and EKPs within high-tech manufacturing organizations. “In short, the evidence suggests that TOP Modeler was successful in supporting organizational design” (p. 187) but additional study is required to assess the comparative effectiveness of other possible approaches in this or other contexts. Again, this is not a criticism of this work; rather it is a call for further research in the general class of problems dealing with emergent knowledge processes. As additional research builds on this foundation, formal, rigorous evaluation and comparison with alternative approaches in a variety of contexts become crucial to enable claims of generalizability. As the authors point out, “Only the accumulated weight of

empirical evidence will establish the validity” of such claims.

Research Contributions

The design-science contributions of this research are the TOP Modeler software and the design principles. TOP Modeler demonstrates the feasibility of using the design principles to develop an artifact to support EKPs. Because TOP Modeler is the first artifact to address this task, its construction is itself a contribution to design science. Furthermore, because the authors are able to articulate the design principles upon which its construction was based, these serve as hypotheses to be tested by future empirical work. Their applicability to the development of other types of information systems can also be tested. An agenda for addressing such issues is presented. This focuses on validation, evaluation, and the challenges of improvement inherent in the evaluation process.

Research Communication

This work presents two types of artifacts, TOP Modeler (an instantiation) and a set of design principles (method) that address a heretofore unsolved problem dealing with the design of an information system to support EKPs. Recognizing that existing system development methods and instantiations are aimed at structured or semi-structured activities, Markus et al. identify an opportunity to apply information technology in a new and innovative way. Their presentation addresses each of the design guidelines posed above. TOP Modeler exemplifies “proof by construction”—it is feasible to construct an information system to support EKPs. Since it is the first such artifact, its evaluation using formal methods is deferred until future research. Technical details of TOP Modeler are not presented, making it difficult for a technical researcher or practitioner to replicate their work. The uniqueness of the artifacts and the innovation inherent in them are presented so that managerial researchers and IT managers are aware of the new capabilities.

Discussion and Conclusions ■

Philosophical debates on how to conduct IS research (e.g., positivism vs. interpretivism) have been the focus of much recent attention (Klein and Myers 1999; Robey 1996; Weber 2003). The major emphasis of such debates lies in the epistemologies of research, the underlying assumption being that of the natural sciences. That is, somewhere some *truth* exists and somehow that truth can be extracted, explicated, and codified. The behavioral-science paradigm seeks to find “what is true.” In contrast, the design-science paradigm seeks to create “what is effective.” While it can be argued that utility relies on truth, the discovery of truth may lag the application of its utility. We argue that both design-science and behavioral-science paradigms are needed to ensure the relevance and effectiveness of IS research. Given the artificial nature of organizations and the information systems that support them, the design-science paradigm can play a significant role in resolving the fundamental dilemmas that have plagued IS research: rigor, relevance, discipline boundaries, behavior, and technology (Lee 2000).

Information systems research lies at the intersection of people, organizations, and technology (Silver et al. 1995). It relies on and contributes to cognitive science, organizational theory, management sciences, and computer science. It is both an organizational and a technical discipline that is concerned with the analysis, construction, deployment, use, evaluation, evolution, and management of information system artifacts in organizational settings (Madnick 1992; Orlikowski and Barley 2001).

Within this setting, the design-science research paradigm is proactive with respect to technology. It focuses on creating and evaluating innovative IT artifacts that enable organizations to address important information-related tasks. The behavioral-science research paradigm is reactive with respect to technology in the sense that it takes technology as “given.” It focuses on developing and justifying theories that explain and predict phenomena related to the acquisition, implemen-

tation, management, and use of such technologies. The dangers of a design-science research paradigm are an overemphasis on the technological artifacts and a failure to maintain an adequate theory base, potentially resulting in well-designed artifacts that are useless in real organizational settings. The dangers of a behavioral-science research paradigm are overemphasis on contextual theories and failure to adequately identify and anticipate technological capabilities, potentially resulting in theories and principles addressing outdated or ineffective technologies. We argue strongly that IS research must be both proactive and reactive with respect to technology. It needs a complete research cycle where design science creates artifacts for specific information problems based on relevant behavioral science theory and behavioral science anticipates and engages the created technology artifacts.

Hence, we reiterate the call made earlier by March et al. (2000) to align IS design-science research with real-world production experience. Results from such industrial experience can be framed in the context of our seven guidelines. These must be assessed not only by IS design-science researchers but also by IS behavioral-science researchers who can validate the organizational problems as well as study and anticipate the impacts of created artifacts. Thus, we encourage collaborative industrial/academic research projects and publications based on such experience. Markus et al. (2002) is an excellent example of such collaboration. Publication of these results will help accelerate the development of domain independent and scalable solutions to large-scale information systems problems within organizations. We recognize that a lag exists between academic research and its adoption in industry. We also recognize the possible *ad hoc* nature of technology-oriented solutions developed in industry. The latter gap can be reduced considerably by developing and framing the industrial solutions based on our proposed guidelines.

It is also important to distinguish between “system building” efforts and design-science research. Guidelines addressing evaluation, contributions, and rigor are especially important in providing this

distinction. The underlying formalism required by these guidelines helps researchers to develop representations of IS problems, solutions, and solution processes that clarify the knowledge produced by the research effort.

As we move forward, there exist a number of exciting challenges facing the design-science research community in IS. A few are summarized here.

- There is an inadequate theoretical base upon which to build an engineering discipline of information systems design (Basili 1996). The field is still very young lacking the cumulative theory development found in other engineering and social-science disciplines. It is important to demonstrate the feasibility and utility of such a theoretical base to a managerial audience that must make technology-adoption decisions that can have far-reaching impacts on the organization.
- Insufficient sets of constructs, models, methods, and tools exist for accurately representing the business/technology environment. Highly abstract representations (e.g., analytical mathematical models) are criticized as having no relationship to “real-world” environments. On the other hand, many informal, descriptive IS models lack an underlying theory base. The trade-offs between relevance and rigor are clearly problematic; finding representational techniques with an acceptable balance between the two is very difficult.
- The existing knowledge base is often insufficient for design purposes and designers must rely on intuition, experience, and trial-and-error methods. A constructed artifact embodies the designer’s knowledge of the problem and solution. In new and emerging applications of technology, the artifact itself represents an experiment. In its execution, we learn about the nature of the problem, the environment, and the possible solutions—hence, the importance of developing and implementing prototype artifacts (Newell and Simon 1976).
- Design-science research is perishable. Rapid advances in technology can invalidate design-science research results before they are implemented effectively in the business environment or, just as importantly to managers, before adequate payback can be achieved by committing organizational resources to implementing those results. Two examples are the promises made by the artificial intelligence community in the 1980s (Feigenbaum and McCorduck 1983) and the more recent research on object-oriented databases (Chaudhri and Loomis 1998). Just as important to IS researchers, design results can be overtaken by technology before they even appear in the research literature. How much research was published on the Year 2000 problem before it became a non-event?
- Rigorous evaluation methods are extremely difficult to apply in design-science research (Tichy 1998; Zelkowitz and Wallace 1998). For example, the use of a design artifact on a single project may not generalize to different environments (Markus et al. 2002).

We believe that design science will play an increasingly important role in the IS profession. IS managers in particular are actively engaged in design activities—the creation, deployment, evaluation, and improvement of purposeful IT artifacts that enable organizations to achieve their goals. The challenge for design-science researchers in IS is to inform managers of the capabilities and impacts of new IT artifacts.

Much of the research published in *MIS Quarterly* employs the behavioral-science paradigm. It is passive with respect to technology, often ignoring or “under-theorizing” the artifact itself (Orlikowski and Iacono 2001). Its focus is on describing the implications of *technology*—its impact on individuals, groups, and organizations. It regularly includes studies that examine how people employ a technology, report on the benefits and difficulties encountered when a technology is implemented within an organization, or discuss how managers might facilitate the use of a technology. Orman (2002) argues that many of the equivocal results

in IS behavioral-science studies can be explained by a failure to differentiate the capabilities and purposes of the studied technology.

Design science is active with respect to technology, engaging in the creation of technological artifacts that impact people and organizations. Its focus is on problem solving but often takes a simplistic view of the people and the organizational contexts in which designed artifacts must function. As stated earlier, the design of an artifact, its formal specification, and an assessment of its utility, often by comparison with competing artifacts, are integral to design-science research. These must be combined with behavioral and organizational theories to develop an understanding of business problems, contexts, solutions, and evaluation approaches adequate to servicing the IS research and practitioner communities. The effective presentation of design-science research in major IS journals, such as *MIS Quarterly*, will be an important step toward integrating the design-science and behavioral-science communities in IS.

Acknowledgements

We would like to thank Allen Lee, Ron Weber, and Gordon Davis who in different ways each contributed to our thinking about design science in the Information Systems profession and encouraged us to pursue this line of research. We would also like to acknowledge the efforts of Rosann Collins who provided insightful comments and perspectives on the nature of the relationship between behavioral-science and design-science research. This work has also benefited from seminars and discussions at Arizona State University, Florida International University, Georgia State University, Michigan State University, Notre Dame University, and The University of Utah. We would particularly like to thank Brian Pentland and Steve Alter for feedback and suggestions they provided on an earlier version of this paper. The comments provided by several anonymous editors and reviewers greatly enhanced the content and presentation of the paper.

References

- Aalst, W. "Wolfan: A Petri-Net-Based Workflow Analyzer," *Systems Analysis-Modeling-Simulation* (34:3), 1999, pp. 345-357.
- Aalst, W., and Hee, K. *Workflow Management: Models, Methods, and Systems*, The MIT Press, Cambridge, MA, 2002.
- Aalst, W., and Kumar, A. "XML-Based Schema Definition for Support of Interorganizational Workflow," *Information Systems Research* (14:1), March 2003, pp. 23-46.
- Aboulafia, M. (ed.). *Philosophy, Social Theory, and the Thought of George Herbert Mead* (SUNY Series in Philosophy of the Social Sciences), State University of New York Press, Albany, NY, 1991.
- Aiken, M. W., Sheng, O. R. L., and Vogel, D. R. "Integrating Expert Systems with Group Decision Support Systems," *ACM Transactions on Information Systems* (9:1), January 1991, pp. 75-95.
- Alter, S. "18 Reasons Why IT-Reliant Work Systems Should Replace 'The IT Artifact' as the Core Subject Matter of the IS Field," *Communications of the AIS* (12), October 2003, pp. 365-394.
- Applegate, L. M. "Rigor and Relevance in MIS Research—Introduction," *MIS Quarterly* (23:1), March 1999, pp. 1-2.
- Basili, V. "The Role of Experimentation in Software Engineering: Past, Current, and Future," in *Proceedings of the 18th International Conference on Software Engineering*, T. Maibaum and M. Zelkowitz (eds.), Berlin, Germany, March 25-29, 1996, pp. 442-449.
- Basu, A., and Blanning, R. W. "A Formal Approach to Workflow Analysis," *Information Systems Research* (11:1), March 2000, pp. 17-36.
- Basu, A., and Blanning, R. W. "Metagraphs: A Tool for Modeling Decision Support Systems," *Management Science* (40:12), December 1994, pp. 1579-1600.
- Batra, D., Hoffer, J. A., and Bostrom, R. P. "A Comparison of User Performance between the Relational and the Extended Entity Relationship Models in the Discovery Phase of Database Design," *Communications of the ACM* (33:2), February 1990, pp. 126-139.

- Bell, D. A. "From Data Properties to Evidence," *IEEE Transactions on Knowledge and Data Engineering* (5:6), December 1993, pp. 965-969.
- Benbasat, I., and Zmud, R. W. "Empirical Research in Information Systems: The Practice of Relevance," *MIS Quarterly* (23:1), March 1999, pp. 3-16.
- Bhargava, H. K., Krishnan, R., and Piela, P. "On Formal Semantics and Analysis of Typed Modeling Languages: An Analysis of Ascend," *INFORMS Journal on Computing* (10:2), Spring 1998, pp. 189-208.
- Bodart, F., Patel, A., Sim, M., and Weber, R. "Should the Optional Property Construct be Used in Conceptual Modeling? A Theory and Three Empirical Tests," *Information Systems Research* (12:4), December 2001, pp. 384-405.
- Boland, R. J. "Design in the Punctuation of Management Action" in *Managing as Designing: Creating a Vocabulary for Management Education and Research*, R. Boland (ed.), Frontiers of Management Workshop, Weatherhead School of Management, June 14-15, 2002 (available online at <http://design.cwru.edu>).
- Brancheau, J., Janz, B., and Wetherbe, J. "Key Issues in Information Systems Management: 1994-95 SIM Delphi Results," *MIS Quarterly* (20:2), June 1996, pp. 225-242.
- Brooks, F. P., Jr. "The Computer Scientist as Toolsmith II," *Communications of the ACM* (39:3), March 1996, pp. 61-68.
- Brooks, F. P., Jr. "No Silver Bullet: Essence and Accidents of Software Engineering," *IEEE Computer* (20:4), April 1987, pp. 10-19.
- Bunge, M. A. *Treatise on Basic Philosophy: Volume 7—Epistemology & Methodology III: Philosophy of Science and Technology—Part II: Life Science, Social Science and Technology*, D. Reidel Publishing Company, Boston, MA, 1985.
- Chaudhri, A., and Loomis, M. *Object Databases in Practice*, Prentice-Hall, Upper Saddle River, NJ, 1998.
- Chaum, D. "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms," *Communications of the ACM* (24:2), February 1981, pp. 84-87.
- Chen, P. P. S. "The Entity-Relationship Model: Toward a Unified View," *ACM Transactions on Database Systems* (1:1), 1976, pp. 9-36.
- Davis, G. B., and Olson, M. H. *Management Information Systems: Conceptual Foundations, Structure and Development* (2nd ed.), McGraw-Hill, New York, 1985.
- DeLone, W. H., and McLean, E. R. "The DeLone and McLean Model of Information Systems Success: A Ten-Year Update," *Journal of Management Information Systems* (19:4), Spring 2003, pp. 9-30.
- DeLone, W. H., and McLean, E. R. "Information Systems Success: The Quest for the Dependent Variable," *Information Systems Research* (3:1), March 1992, pp. 60-95.
- Denning, P. J. "A New Social Contract for Research," *Communications of the ACM* (40:2), February 1997, pp. 132-134.
- Dennis, A., George, J., Jessup, L., Nunamaker, J., and Vogel, D. "Information Technology to Support Electronic Meetings," *MIS Quarterly* (12:4), December 1988, pp. 591-624.
- Dennis, A., and Wixom, B. "Investigating the Moderators of the Group Support Systems Use with Meta-Analysis," *Journal of Management Information Systems* (18:3), Winter 2001-02, pp. 235-257.
- Dey, D., Sarkar, S., and De, P. "A Probabilistic Decision Model for Entity Matching in Heterogeneous Databases," *Management Science* (44:10), October 1998, pp. 1379-1395.
- Dey, D., Storey, V. C., and Barron, T. M. "Improving Database Design through the Analysis of Relationships," *ACM Transactions on Database Systems* (24:4), December 1999, pp. 453-486.
- Dickson, G., Partridge, J., and Robinson, L. "Exploring Modes of Facilitative Support for GDSS Technology," *MIS Quarterly* (17:2), June 1993, pp. 173-194.
- Drucker, P. F. "The Coming of the New Organization," *Harvard Business Review* (66:1), January-February 1988, pp. 45-53.
- Drucker, P. F. "The New Productivity Challenge," *Harvard Business Review* (69:6), November-December 1991, pp. 45-53.
- Dym, C. L. *Engineering Design*, Cambridge University Press, New York, 1994.

- eMarketer. *E-Commerce Trade and B2B Exchanges*, March 2002 (available online at http://www.emarketer.com/products/report.php?ecommerce_trade).
- Feigenbaum, E., and McCorduck, P. *The Fifth Generation: Artificial Intelligence and Japan's Computer Challenge to the World*, Addison-Wesley, Inc., Reading, MA, 1983.
- Fjermestad, J., and Hiltz, S. R. "An Assessment of Group Support Systems Experimental Research: Methodology and Results," *Journal of Management Information Systems* (15:3), Winter 1998-99, pp. 7-149.
- Gallupe, R., DeSanctis, G., and Dickson, G. "Computer-Based Support for Group Problem-Finding: An Experimental Investigation," *MIS Quarterly*, (12:2), June 1988, pp. 277-298.
- Gavish, B., and Gerdes, J. "Anonymous Mechanisms in Group Decision Support Systems Communication," *Decision Support Systems* (23:4), October 1998, pp. 297-328.
- Geerts, G., and McCarthy, W. E. "An Ontological Analysis of the Primitives of the Extended-REA Enterprise Information Architecture," *The International Journal of Accounting Information Systems* (3:1), 2002, pp. 1-16.
- Gelernter, D. *Machine Beauty: Elegance and the Heart of Technology*, Basic Books, New York, 1998.
- Glass, R. "On Design," *IEEE Software* (16:2), March/April 1999, pp. 103-104.
- Halpin, T. A. *Information Modeling and Relational Databases*, Morgan Kaufmann Publishers, New York, 2001.
- Henderson, J., and Venkatraman, N. "Strategic Alignment: Leveraging Information Technology for Transforming Organizations," *IBM Systems Journal* (32:1), 1993.
- ISR. Editorial Statement and Policy, *Information Systems Research* (13:4), December 2002.
- Jarvenpaa, S., Rao, V., and Huber, G. "Computer Support for Meetings of Groups Working on Unstructured Problems: A Field Experiment," *MIS Quarterly* (12:4), December 1988, pp. 645-666.
- Johansson, J. M. "On the Impact of Network Latency on Distributed Systems Design," *Information Technology Management* (1), 2000, pp. 183-194.
- Johansson, J. M., March, S. T., and Naumann, J. D. "Modeling Network Latency and Parallel Processing in Distributed Database Design," *Decision Sciences Journal* (34:4), Fall 2003.
- Kalakota, R., and Robinson, M. *E-Business 2.0: Roadmap for Success*, Addison-Wesley Pearson Education, Boston, MA, 2001.
- Keil, M. "Pulling the Plug: Software Project Management and the Problem of Project Escalation," *MIS Quarterly* (19:4) December 1995, pp. 421-447.
- Keil, M., Cule, P. E., Lyytinen, K., and Schmidt, R. C. "A Framework for Identifying Software Project Risks," *Communications of the ACM* (41:11), November 1998, pp. 76-83.
- Keil, M., and Robey, D. "Turning Around Troubled Software Projects: An Exploratory Study of the Deescalation of Commitment to Failing Courses of Action," *Journal of Management Information Systems*, (15:4) December 1999, pp. 63-87.
- Kernighan, B., and Plauger, P. J. *The Elements of Programming Style* (2nd ed.), McGraw-Hill, New York, 1978.
- Kim, Y. G., and March, S. T. "Comparing Data Modeling Formalisms," *Communications of the ACM* (38:6), June 1995, pp. 103-115.
- Klein, H. K., and Myers, M. D. "A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems," *MIS Quarterly* (23:1), March 1999, pp. 67-94.
- Kleindorfer, G., O'Neill, L., and Ganeshan, R. "Validation in Simulation: Various Positions in the Philosophy of Science," *Management Science* (44:8), August 1998, pp. 1087-1099.
- Kruchten, P. *The Rational Unified Process: An Introduction* (2nd ed.), Addison-Wesley, Inc., Reading, MA, 2000.
- Kuhn, T. S. *The Structure of Scientific Revolutions* (3rd ed.), University of Chicago Press, Chicago, IL, 1996.
- Kumar, A., and Zhao, J. "Workflow Support for Electronic Commerce Applications," *Decision Support Systems* (32:3), January 2002, pp. 265-278.
- Lee, A. "Inaugural Editor's Comments," *MIS Quarterly* (23:1), March 1999, pp. v-xi.
- Lee, A. "Systems Thinking, Design Science, and Paradigms: Heeding Three Lessons from the Past to Resolve Three Dilemmas in the

- Present to Direct a Trajectory for Future Research in the Information Systems Field," Keynote Address, Eleventh International Conference on Information Management, Taiwan, May 2000 (available online at <http://www.people.vcu.edu/~aslee/ICIM-keynote-2000>).
- Leymann, F., and Roller, D. *Production Workflow: Concepts and Techniques*, Prentice-Hall, Upper Saddle River, NJ, 2000.
- Madnick, S. E. "The Challenge: To Be Part of the Solution Instead of Being Part of the Problem," in *Proceedings of the Second Annual Workshop on Information Technology and Systems*, V. Storey and A. Whinston (eds.), Dallas, TX, December 12-13, 1992, pp. 1-9.
- Marakas, G. M., and Elam, J. J. "Semantic Structuring in Analyst Acquisition and Representation of Facts in Requirements Analysis," *Information Systems Research* (9:1), March 1998, pp. 37-63.
- March, S. T., Hevner, A., and Ram, S. "Research Commentary: An Agenda for Information Technology Research in Heterogeneous and Distributed Environment," *Information Systems Research* (11:4), December 2000, pp. 327-341.
- March, S. T., and Smith, G. "Design and Natural Science Research on Information Technology," *Decision Support Systems* (15:4), December 1995, pp. 251-266.
- Markus, M. L., Majchrzak, A., and Gasser, L. "A Design Theory for Systems that Support Emergent Knowledge Processes," *MIS Quarterly* (26:3), September, 2002, pp. 179-212.
- McCarthy, W. E. "The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment," *The Accounting Review* (58:3), 1982, pp. 554-578.
- Meehl, P. E. "What Social Scientists Don't Understand," in *Metatheory in Social Science*, D. W. Fiske and R. A. Shweder (eds.), University of Chicago Press, Chicago, IL, 1986, pp. 315-338.
- Newell, A., and Simon, H. "Computer Science as Empirical Inquiry: Symbols and Search," *Communications of the ACM* (19:3), March 1976, pp. 113-126.
- Norman, D. *The Design of Everyday Things*, Currency Doubleday, New York, 1988.
- Nunamaker, J., Briggs, R., Mittleman, D., Vogel, D., and Balthazard, P. "Lessons from a Dozen Years of Group Support Systems Research: A Discussion of Lab and Field Findings," *Journal of Management Information Systems*, (13:3), Winter 1996-97, pp. 163-207.
- Nunamaker, J., Chen, M., and Purdin, T. D. M. "Systems Development in Information Systems Research," *Journal of Management Information Systems* (7:3), Winter 1991a, pp. 89-106.
- Nunamaker, J., Dennis, A., Valacich, J., Vogel, D., and George, J. "Electronic Meeting Systems to Support Group Work," *Communications of the ACM*, (34:7), July 1991b, pp. 40-61.
- Orlikowski, W. J. "Using Technology and Constituting Structures: A Practice Lens for Studying Technology in Organizations." *Organization Science* (11:4), December 2000, pp. 404-428.
- Orlikowski, W. J., and Barley, S. R. "Technology and Institutions: What Can Research on Information Technology and Research on Organizations Learn From Each Other?," *MIS Quarterly* (25:2), June 2001, pp 145-165.
- 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), June 2001, pp. 121-134.
- Orman, L. V. "Electronic Markets, Hierarchies, Hubs, and Intermediaries," *Journal of Information Systems Frontiers* (4:2), 2002, pp. 207-222.
- Pahl, G., and Beitz, W. *Engineering Design: A Systematic Approach*, Springer-Verlag, London, 1996.
- Parsons, J., and Wand, Y. "Emancipating Instances from the Tyranny of Classes in Information Modeling," *ACM Transactions on Database Systems* (25:2), June 2000, pp. 228-268.
- Petroski, H. *Invention by Design: How Engineers Get from Thought to Thing*, Harvard University Press, Cambridge, MA, 1996.

- Purao, S., Storey, V. C., and Han, T. D. "Improving Reuse-Based System Design with Learning," *Information Systems Research* (14:3), September 2003, pp. 269-290.
- Rittel, H. J., and Webber, M. M. "Planning Problems Are Wicked Problems," in *Developments in Design Methodology*, N. Cross (ed.), John Wiley & Sons, New York, 1984.
- Robey, D. "Research Commentary: Diversity in Information Systems Research: Threat, Opportunity, and Responsibility," *Information Systems Research* (7:4), 1996, pp. 400-408.
- Royce, W. *Software Project Management: A Unified Framework*, Addison-Wesley, Inc., Reading, MA, 1998.
- Salton, G. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*, Addison-Wesley, Inc., Reading, MA, 1988.
- Schneier, B. *Applied Cryptography: Protocols, Algorithms, and Source Code in C* (2nd ed.), John Wiley and Sons, New York, January 1996.
- Schön, D. A. *The Reflective Practitioner: How Professionals Think in Action*, Basic Books, New York, 1983.
- Seddon, P. B. "A Respecification and Extension of the DeLone and McLean Model of IS Success," *Information Systems Research* (8:3), September 1997, pp. 240-253.
- Sengupta, K., and Te'eni, D. "Cognitive Feedback in GDSS: Improving Control and Convergence," *MIS Quarterly* (17:1), March 1993, pp. 87-113.
- Silver, M. S., Markus, M. L., and Beath, C. M. "The Information Technology Interaction Model: A Foundation for the MBA Core Course," *MIS Quarterly* (19:3), September 1995, pp. 361-390.
- Simon, H. A. *The Sciences of the Artificial* (3rd ed.), MIT Press, Cambridge, MA, 1996.
- Sinha, A. P., and Vessey, I. "An Empirical Investigation of Entity-Based and Object-Oriented Data Modeling: A Development Life Cycle Approach," in *Proceedings of the Twentieth International Conference on Information Systems*, P. De and J. I. DeGross (eds.), Charlotte, NC, December 13-15, 1999, pp. 229-244.
- Storey, V. C., Chiang, R. H. L., Dey, D., Goldstein, R. C., and Sundaresan, S. "Database Design with Common Sense Business Reasoning and Learning," *ACM Transactions on Database Systems* (22:4), December 1997, pp. 471-512.
- Tam, K. Y. "Automated Construction of Knowledge-Bases from Examples," *Information Systems Research* (1:2), June 1990, pp. 144-167.
- Tichy, W. "Should Computer Scientists Experiment More?" *IEEE Computer* (31:5), May 1998, pp. 32-40.
- Trice, A., and Davis, R. "Heuristics for Reconciling Independent Knowledge Bases," *Information Systems Research* (4:3), September 1993, pp. 262-288.
- Tsichritzis, D. "The Dynamics of Innovation," in *Beyond Calculation: The Next Fifty Years of Computing*, P. J. Denning and R. M. Metcalfe (eds.), Copernicus Books, New York, 1998, pp. 259-265.
- Venkatesh, V. "Determinants of Perceived Ease of Use: Integrating Control, Intrinsic Motivation, and Emotion into the Technology Acceptance Model," *Information Systems Research* (11:4), December 2000, pp. 342-365.
- Vessey, I., and Glass, R. "Strong Vs. Weak Approaches to Systems Development," *Communications of the ACM* (41:4), April 1998, pp. 99-102.
- Walls, J. G., Widmeyer, G. R., and El Sawy, O. A. "Building an Information System Design Theory for Vigilant EIS," *Information Systems Research* (3:1), March 1992, pp. 36-59.
- Wand, Y., and Weber, R. "On the Deep Structure of Information Systems," *Information Systems Journal* (5), 1995, pp. 203-233.
- Wand, Y., and Weber, R. "On the Ontological Expressiveness of Information Systems Design Analysis and Design Grammars," *Journal of Information Systems* (3:3), 1993, pp. 217-237.
- Weber, R. "Editor's Comments: Still Desperately Seeking the IT Artifact," *MIS Quarterly* (27:2), June 2003, pp. iii-xi.
- Weber, R. *Ontological Foundations of Information Systems*, Coopers & Lybrand, Brisbane, Australia, 1997.

- Weber, R. "Toward a Theory of Artifacts: A Paradigmatic Base for Information Systems Research," *Journal of Information Systems* (1:2), Spring 1987, pp. 3-19.
- WfMC. "Workflow Standard—Interoperability WfXML Binding," Document Number WfMC-TC-1023, Version 1.0, Workflow Management Coalition, 2000.
- Winograd, T. *Bringing Design to Software*, Addison-Wesley, Inc., Reading, MA, 1996.
- Winograd, T. "The Design of Interaction," in *Beyond Calculation: The Next 50 Years of Computing*, P. Denning and R. Metcalfe (eds.), Copernicus Books, New York, 1998, pp. 149-162.
- Zelkowitz, M., and Wallace, D. "Experimental Models for Validating Technology," *IEEE Computer* (31:5), May 1998, pp. 23-31.
- Zmud, R. "Editor's Comments," *MIS Quarterly* (21:2), June 1997, pp. xxi-xxii.

About the Authors

Alan R. Hevner is an Eminent Scholar and Professor in the College of Business Administration at the University of South Florida. He holds the Salomon Brothers/Hidden River Corporate Park Chair of Distributed Technology. His areas of research interest include information systems development, software engineering, distributed database systems, and healthcare information systems. He has published numerous research papers on these topics and has consulted for several Fortune 500 companies. Dr. Hevner received a Ph.D. in Computer Science from Purdue University. He has held faculty positions at the University of Maryland at College Park and the University of Minnesota. Dr. Hevner is a member of ACM, IEEE, AIS, and INFORMS.

Salvatore T. March is the David K. Wilson Professor of Management at the Owen Graduate School of Management, Vanderbilt University. He received a B.S. in Industrial Engineering and M.S. and Ph.D. degrees in Operations Research from Cornell University. His research interests are in information system development, distributed data-

base design, and electronic commerce. His research has appeared in journals such as *Communications of the ACM*, *IEEE Transactions on Knowledge and Data Engineering*, and *Information Systems Research*. He served as the Editor-in-Chief of *ACM Computing Surveys* and as an associate editor for *MIS Quarterly*. He is currently a senior editor for *Information Systems Research* and an associate editor for *Decision Sciences Journal*.

Jinsoo Park is an assistant professor of information systems in the College of Business Administration at Korea University. He was formerly on the faculty of the Carlson School of Management at the University of Minnesota. He holds a Ph.D. in MIS from the University of Arizona. His research interests are in the areas of semantic interoperability and metadata management in interorganizational information systems, heterogeneous information resource management and integration, knowledge sharing and coordination, and data modeling. His published research articles appear in *IEEE Computer*, *IEEE Transactions on Knowledge and Data Engineering*, and *Information Systems Frontiers*. He currently serves on the editorial board of *Journal of Database Management*. He is a member of ACM, IEEE, AIS, and INFORMS.

Sudha Ram is the Eller Professor of MIS at the University of Arizona. She received a B.S. in Science from the University of Madras in 1979, PGDM from the Indian Institute of Management, Calcutta in 1981 and a Ph.D. from the University of Illinois at Urbana-Champaign, in 1985. Dr. Ram has published articles in such journals as *Communications of the ACM*, *IEEE Transactions on Knowledge and Data Engineering*, *Information Systems Research*, and *Management Science*. Her research deals with interoperability in heterogeneous databases, semantic modeling, data allocation, and intelligent agents for data management. Her research has been funded by IBM, National Institute of Standards and Technology (NIST), National Science Foundation (NSF), National Aeronautics and Space Administration (NASA), and the Office of Research and Development of the Central Intelligence Agency (CIA).

Appendix B

Exemplar Publications of Design Science Research in Information Systems

The design science research paradigm is poised to take its rightful place as a synergistic and equal partner alongside other research paradigms in the information systems (IS) field. In doing so, it is important to recognize some of the seminal design science research that has appeared in the IS literature up to the time of this text. The following list of exemplar publications, while in no means complete, provides a starting point for exploring some of the most significant design science contributions of the IS field over the past 30 years. The goals of this appendix are to highlight recent thinking on design science theories and research practices and to provide exemplar design science papers to contribute to a greater understanding of design science research in the IS community. Communicating IS design science theories and research practices is essential not only to support acceptance among IS professionals but also to establish the credibility of IS design science research among the larger body of design science researchers in computer science, engineering fields, architecture, the arts, and other design-oriented communities.

These exemplar papers are grouped into categories based on research topics. Volume III in the *SAGE Library in Business & Management* provides a collection of a subset of these exemplar papers (Hevner 2008 – Number 5 in the list below).

Design Science Theory and Practice

1. Gregor, S. and D. Jones (2007) The anatomy of a design theory, *Journal of the AIS* 8 (5), Article 2, pp. 312–335.
2. Hevner, A., S. March, J. Park, and S. Ram (2004) Design science in information systems research, *MIS Quarterly* 28 (1), pp. 75–105.
3. Hevner, A. (2007) A three-cycle view of design science research, *Scandinavian Journal of Information Systems* 19 (2), pp. 87–92.
4. Peffers K., T. Tuunanen, M. A. Rothenberger, and S. Chatterjee (2007-8) A design science research methodology for information systems research, *Journal of Management Information Systems (JMIS)* 24 (3), pp. 45–77.
5. Hevner, A. (Volume Editor) (2008) Design science theories and research practices, in L. Willcocks and A. Lee (eds.) Volume III in the *SAGE Major Currents in Information Systems Series*, SAGE Publications, Inc., Thousand Oaks, CA.
6. Iivari, J. (1991) A paradigmatic analysis of contemporary schools of IS development, *European Journal of Information Systems* 1 (4), pp. 249–272.

7. Iivari, J. (2007) A paradigmatic analysis of information systems as a design science, *Scandinavian Journal of IS* 19 (2), pp. 39–64.
8. March, S. and G. Smith (1995) Design and natural science research on information technology, *Decision Support Systems* 15 (4), pp. 251–266.
9. Markus, M., A. Majchrzak, L. Gasser (2002) A design theory for systems that support emergent knowledge processes, *MIS Quarterly* 26 (3), pp. 179–212.
10. Nunamaker, J., M. Chen, and T. Purdin (1991) Systems development in information systems research, *Journal of Management Information Systems* 7 (3), pp. 89–106.
11. Purao, S., C. Baldwin, A. Hevner, V. Storey, J. Pries-Heje, B. Smith, and Y. Zhu (2008) The sciences of design: observations on an emerging field, *Communications of the AIS*, 23, Article 29, pp. 523–546.
12. Peffers, K., T. Tuunanen, M. Rothenberger, and S. Chatterjee (2008) A design science research methodology for information systems research, *Journal of Management Information Systems*, 24 (3), pp. 45–77.
13. Walls, J., G. Widmeyer, and O. El Sawy (1992) Building an information system design theory for vigilant EIS, *Information Systems Research* 3 (1), pp. 36–59.
14. Weber, R. (1987) Towards a theory of artifacts: a paradigmatic base for information systems research, *Journal of Information Systems* 1 (1), pp. 3–20.

Information Systems Development and Systems Modeling

15. Abdel-Hamid, T. and S. Madnick (1989) Lessons learned from modeling the dynamics of software development, *Communications of the ACM* 32 (12), pp. 1426–1438, 1455.
16. Chidamber, S. and C. Kemerer (1994) A metrics suite for object oriented design, *IEEE Transactions on Software Engineering* 20 (6), 476–493.
17. Freeman, P. and D. Hart (2004) A science of design for software-intensive systems, *Communications of the ACM* 47 (8), pp. 19–21.
18. Hevner, A. and H. Mills (1993) Box structured methods for systems development with objects, *IBM Systems Journal* 32 (2), pp. 232–251.

Information and Data Modeling

19. Chen, P. (1976) The entity-relationship model: toward a unified model of data, *ACM Transactions on Database Systems* 1 (1), pp. 9–36.
20. Parsons, J. and Y. Wand (2000) Emancipating instances from the tyranny of classes in information modeling, *ACM Transactions on Database Systems* 25 (2), pp. 228–268.
21. Ram, S. and V. Khatri (2005) A comprehensive framework for modeling set-based business rules during conceptual database design, *Information Systems*, 30 (2), pp. 89–118.
22. Ram, S. and J. Park (2004) Semantic conflict resolution ontology (SCROL): an ontology for detecting and resolving data and schema-level semantic conflicts, *IEEE Transactions on Knowledge and Data Engineering* 16 (2), pp. 189–202.

23. Wand, Y. and R. Weber (1990) An ontological model of an information system, *IEEE Transactions on Software Engineering* 16 (11), pp. 1282–1292.
24. Wand, Y., V. Storey, and R. Weber, An ontological analysis of the relationship construct in conceptual modeling, *ACM Transactions on Database Systems* 24 (4), pp. 494–528.
25. Wang, R., V. Storey, and C. Firth (1995) a framework for analysis of data quality research, *IEEE Transactions on Knowledge and Data Engineering* 7 (4), pp. 623–640.

Database Systems Design

26. Dey, D. and S. Sarkar (1996) A probabilistic relational model and algebra, *ACM Transactions on Database Systems* 21 (3), pp. 339–369.
27. Ram, S. and S. Narasimhan (1994) Database allocation in a distributed environment: incorporating a concurrency control mechanism and queueing costs, *Management Science* 40 (8), pp. 969–983.
28. Storey, V., R. Chiang, D. Dey, R. Goldstein, and S. Sundaresan, Database design with common sense business reasoning and learning, *ACM Transactions on Database Systems* 22 (4), pp. 471–512.

Knowledge and Information Integration

29. Goh, C., S. Bressan, S. Madnick, and M. Siegel (1999) context interchange: new features and formalisms for the intelligent integration of information, *ACM Transactions on Information Systems* 17 (3), pp. 270–293.
30. Krishnan, R., X. Li, D. Steier, and J. Zhao (2001) On heterogeneous database retrieval: a cognitively-guided approach, *Information Systems Research* 12 (3), pp. 286–303.
31. Mylopoulos, J., A. Borgida, M. Jarke, and M. Koubarakis (1990) Telos: representing knowledge about information systems, *ACM Transactions on Information Systems* 8 (4), pp. 325–362.
32. Park, J. and S. Ram (2004) Information systems interoperability: what lies beneath? *ACM Transactions on Information Systems* 22 (4), pp. 595–632.

Data Warehousing and Mining

33. Berndt, D., A. Hevner, and J. Studnicki (2003) The CATCH data warehouse: support for community health care decision making, *Decision Support Systems* 35, pp. 367–384.
34. Datta, A., D. VanderMeer, and K. Ramamritham (2002) Parallel star join + data indexes: efficient query processing in data warehouses and OLAP, *IEEE Transactions on Knowledge and Data Engineering* 14 (6), pp. 1299–1316.
35. Silberschatz, A. and A. Tuzhilin (1996) What makes patterns interesting in knowledge discovery systems, *IEEE Transactions on Knowledge and Data Engineering*, 8 (6), pp. 970–974.

Network and Telecommunications Systems

36. Chatterjee, S., T. Abhichandani, B. Tulu, and H. Li (2005) SIP-based enterprise converged network for voice/video over IP: implementation and evaluation of components, *IEEE Journal on Selected Areas in Communications*, 23 (10), pp. 1921–1933.

Decision Support Systems

37. Adomavicius, G., R. Shankaranarayanan, S. Sen, and A. Tuzhilin (2005) Incorporating contextual information in recommender systems using a multidimensional approach, *ACM Transactions on Information Systems* 23 (1), pp. 103–145.
38. Basu, A. and R. Blanning (1994) Metagraphs: a tool for modeling decision support systems, *Management Science* 40 (12), pp. 1579–1600.
39. Nunamaker, J., A. Dennis, J. Valacich, D. Vogel, and J. George (1991) Electronic meeting systems, *Communications of the ACM* 34 (7), pp. 40–61.
40. Nunamaker, J., R. Briggs, D. Mittleman, and D. Vogel (1997) Lessons from a dozen years of group support systems research: a discussion of lab and field findings, *Journal of Management Information Systems* 13 (3), pp. 163–207.

Workflow Systems

41. Kumar, A. and J. Zhao (1999) Dynamic routing and operational controls in workflow management systems, *Management Science* 45 (2), pp 253–272.
42. van der Aalst, W. and A. Kumar (2003) XML-based schema definition for support of interorganizational workflow, *Information Systems Research* 14 (1), pp. 23–46.

Electronic Commerce Systems

43. Bapna, R., P. Goes, and A. Gupta (2003) Analysis and design of business-to-consumer online auctions, *Management Science* 49 (1), pp. 85–101.
44. Chen, H., A. Houston, R. Sewell, and B. Schatz (1998) Internet browsing and searching: user evaluations of category map and concept space techniques, *Journal of the American Society for Information Science* 49 (7), pp. 582–603.

Contributors

Donald J. Berndt (dberndt@coba.usf.edu) is an associate professor in the Information Systems and Decision Sciences Department in the College of Business at the University of South Florida. His research interests include data warehousing, knowledge discovery, data mining, and health informatics. Dr. Berndt received a Ph.D. in information systems from the Stern School of Business at New York University. He was a research scientist at Yale University and Scientific Computing Associates, where he participated in the development of commercial versions of the Linda parallel-programming environment. He also developed artificial intelligence applications in academic settings and at Cognitive Systems, Inc. Dr. Berndt is a member of Beta Gamma Sigma, AAAI, ACM, and AIS.

Sven A. Carlsson (Sven_Carlsson@hermes.ics.lu.se) is professor of informatics at School of Economics and Management, Lund University. His current research interests include the use of IS to support management processes, knowledge management, enterprise systems, technochange, design and redesign of e-business processes in electronic value chains, and networks in turbulent and high-velocity environments. He has a keen interest in the use of critical realism in IS research and IS design science research. He has held visiting positions at universities in Europe, Australia, the USA, and Singapore. He is a regional editor for *Knowledge Management Research and Practice*. He has published more than 100 peer-reviewed journal articles, book chapters, and conference papers and his work has appeared in journals like *JMIS*, *Decision Sciences*, *Information & Management*, *International Journal of Technology Management*, *Knowledge and Process Management*, *Knowledge Management Research & Practice*, *Information Systems and e-Business Management*, and *Scandinavian Journal of IS*.

Juhani Iivari (juhani.iivari@oulu.fi) is a professor in information systems at the University of Oulu, Finland. He is a senior editor of *MIS Quarterly* and serves in editorial boards of four other IS journals. His research has broadly focused on theoretical foundations of information systems, design science research in information systems, information systems development methodologies and approaches, organizational analysis, implementation and acceptance of information systems, and the quality of information systems. He has published in journals such as

Communications of the ACM, Data Base, European Journal of Information Systems, Information & Management, Information Systems, Information Systems Journal, Information Systems Research, Journal of Management Information Systems, Journal of Organizational Computing and Electronic Commerce, MIS Quarterly, and Omega.

Robert Judge (rjudge@mail.sdsu.edu) holds undergraduate degrees in biology and botany, an MBA, and a PhD in the Management of Information Systems and Technology from Claremont Graduate University. His career spans the semiconductor, aerospace, consumer electronics, and Internet service industries at mid-management and executive levels. He has held functional responsibilities that include materiel, manufacturing, information systems, marketing, project management, and customer support. Throughout the last 20+ years of his career, he has served both San Diego State University and the University of San Diego as an adjunct professor teaching graduate and undergraduate courses in operations, supply chain management, manufacturing planning and control systems, project management, and information systems. His current research interests lie in understanding how barriers to knowledge flow arise as small organizations grow, and also in how knowledge management systems in supply chains can influence innovation and the flow of non-logistical knowledge.

Salvatore T. March (sal.march@owen.vanderbilt.edu) is the David K. Wilson Professor of Management at the Owen Graduate School of Management, Vanderbilt University. He received his B.S., M.S., and Ph.D. degrees in operations research from Cornell University. His primary teaching responsibilities and research interests are in the areas of data representation and analysis formalisms, design as a research paradigm, and information system development tools and methodologies. His research has appeared in journals such as *ACM Computing Surveys, ACM Transactions on Database Systems, Communications of the ACM, IEEE Transactions on Knowledge and Data Engineering, Information Systems Research, MIS Quarterly, and The Journal of MIS*. He has served as the editor-in-chief for *ACM Computing Surveys*, senior editor for *Information Systems Research*, and associate editor for *MIS Quarterly*. He is currently an associate editor for *Decision Sciences Journal, Journal of Database Management, and Communications of the AIS*.

Sandeep Purao (spurao@ist.psu.edu) is an associate professor of Information Sciences and Technology at Penn State. Prior to joining Penn State, Purao was on the faculty at Georgia State University. Purao's research focuses on the design, evolution, and management of complex techno-organizational systems. His current projects include process-focused composition and monitoring of service-based systems, risk mitigation in large-scale organizational IT integration projects, and investigation of web service standardization processes. Purao's work has been published in journals such as *Communications of the ACM, IEEE Transactions on Systems, Man and Cybernetics-A, ACM Computing Surveys, and Information Systems Research*; and conferences such as the International Conference on

Information Systems, IEEE Service-Oriented Computing Conference, and the International Conference on Conceptual Modeling. He currently serves as an associate editor for *MIS Quarterly*. Puroo holds a Ph.D. in management information systems from the University of Wisconsin-Milwaukee. He is a member of AIS, ACM, and IEEE.

Matti Rossi (matti.rossi@iki.fi) is a professor of information systems at Helsinki School of Economics. He has worked as research fellow at Erasmus University Rotterdam, visiting assistant professor at Georgia State University, Atlanta, and visiting professor at Claremont Graduate University. He received his Ph.D. in Business Administration from the University of Jyväskylä in 1998. He has been the principal investigator in several major research projects funded by the technological development center of Finland and Academy of Finland. His research papers have appeared in journals such as *CACM*, *Journal of AIS*, *Information and Management* and *Information Systems*, and over 30 of them have appeared in conferences such as ICIS, HICSS, and CAiSE.

Maung K. Sein (maung.k.sein@uia.no) is a professor of information systems at University of Agder, Norway. A Ph.D. from Indiana University, he has led a nomadic life having previously served at Georgia State, Florida International, and Indiana Universities in USA and as an adjunct professor at the University of Bergen in Norway. His research interests have moved on from end-user training and cognitive issues in IS to his current focus on IT and national development, e-government, and proactive research approaches, specifically design research and action research and the relationship between the two. He has published in, among others, *Information Systems Research*, *MIS Quarterly*, *Communications of the ACM*, and *Human-Computer Interaction* and has widely presented his research. His editorial board experience includes *MIS Quarterly*, *MIS Quarterly Executive*, *Communications of the AIS*, *e-Service Journal*, *Electronic Journal of e-Government*, *IT for Development Journal* and guest-editing special issues of *The Communications of the ACM* and *Scandinavian Journal of Information Systems*. He has chaired and served on the program committees of several conferences. He has conducted workshops and seminars on research methods, particularly focusing on design and action research in several countries around the world.

Monica Chiarini Tremblay (mtremblay68@gmail.com) is an assistant professor at Florida International University in Miami, FL. Her research interests are in the areas of data warehouse, data mining, and business intelligence, particularly in the area of health care. Dr. Tremblay has published papers in *Decision Support Systems*, *The Journal of Computer Information Systems* and has presented her research at several international conferences. She is currently working as a co-investigator in two veterans administration, health services research, and development funded studies: "Using Knowledge Discovery Strategies to Identify Fall-related Injuries in the VA", and "Using Text Mining to Differentiate Between Post Traumatic Stress Disorder and Mild Traumatic Brain Injury in Operations Iraqi Freedom and

Enduring Freedom Veterans.” She teaches business intelligence at both the graduate and undergraduate levels and will introduce a new graduate course in health information management in the spring of 2010. Prior to joining academia, Monica worked for Exxon U.S.A., IBM, and Siemens.

Timothy J. Vogus (timothy.vogus@owen.vanderbilt.edu) is assistant professor of management at the Vanderbilt Owen Graduate School of Management. He received his Ph.D. in management and organizations from the Ross School of Business at the University of Michigan. His research has focused on how micro-level actions build, grow, and sustain organizational capabilities, and how organizations can be designed and structured to foster these actions and capabilities. This research has resulted in theories of two critical capabilities: (1) mindful organizing – the capability to detect and correct emerging errors and other unexpected events and (2) resilience – the capability to achieve desirable outcomes amid adversity, strain, and significant barriers to adaptation or development. His research has been published in both managerial and health services outlets including the *Journal of Organizational Behavior* and *Medical Care*. He is currently a member of the editorial board for *Organization Science*.

Kevin Williams (kevin.williams@cgu.edu) is a Ph.D. student in the School of Information Systems and Technology at Claremont Graduate University, California. He holds a B.S. in business administration and an M.S. in finance (M.B.A.) both from California State University, Long Beach. Since 1991 he has been working with mission critical databases on large UNIX systems, first as an employee and for the past 13 years as an independent consultant for a variety of world-class organizations. He has founded two successful consulting companies Gibraltar Associates, Inc. (1996) and Avalon Consulting Solutions, Inc. (2005).

Index

A

- Action/design research, integration
 - AR and DR, arguments, 189–190
 - adding reflection to augment learning, 190–191
 - concretizing learning by adding “Build,” 191
 - integrated research process, envisioning, 191–192
 - criteria, cross-application
 - action research criteria to design research exemplar, 183–186
 - design research criteria to action research exemplar, 187–189
 - research approaches
 - action research, 182–183
 - design research, 180–181
- Action research
 - canonical action research criteria, 183t
 - five-phase cyclical process, 182–183
 - goals, 182
- Action research criteria to design research exemplar
 - cyclical process model, 184
 - principle of change through action, 185–186
 - client motivation, 185
 - principle of learning through reflection, 186
 - applications, 186t
 - principle of theory, 185
 - researcher–client agreement, 184
- Agile development methods, 104
- ALGOL-68, 94
- Analytical modeling, 114–115, 119
- Andrew Sayer’s “*Method in Social Science*”, 215
- Anthropological theory, 195
- Anthropology, 81–82
- AO, *see* Aspect-oriented (AO) design

- Apple, 7, 82, 99, 153, 236, 239, 246, 251
- Apple’s iTunes.com, 246
- Applied research, 3, 45, 71
- Aspect-oriented (AO) design, 94

B

- Baan, 92
- Basic fundamental research, 3
- Batch sequential architecture, *see* Data-flow architecture
- Before–after study, 117
- Behavioral science research, 11f, 13, 50, 121–122, 188, 205, 271, 281–282, 285
- Behavior/behavioral sciences, 5, 10–13, 50, 121–123, 181, 188, 205, 216–217, 223
- Benchmarking idea, 138
- Bernoulli distribution, 163
- Blocks
 - cultural, 149
 - emotional, 148
 - environmental, 149
 - intellectual/expressive, 149–150
 - perceptual, 148–149
- Blueprint, *see* Mapping

C

- CAM, *see* Computer-aided manufacturing (CAM)
- Capability maturity model (CMM), 101
 - principal goal, 102
 - process maturity, levels, 101
 - ad hoc (initial), 101
 - effective (defined), 101
 - mature (managed), 101
 - optimized (optimized), 101
 - stable (repeatable), 101

- Causal theory, 36
 - CICS operating system, 99
 - CIT, *see* Computing and information technology (CIT)
 - C language, 94–95
 - C++ language, 94–95
 - Cleanroom method, 103
 - Cloak of theoretical legitimacy, 39
 - CMM, *see* Capability maturity model (CMM)
 - COBOL (common business-oriented language), 89, 93–95, 102
 - Code and fix process, 101
 - Cohen's kappa, 137
 - Commercial off-the-shelf (COTS), 92, 219
 - Complex systems, 24–25, 65–66, 69, 71, 73–74, 88
 - Complex systems-of-systems architecture, 66
 - Component-based architecture
 - basic perspectives, 91
 - competitive potential, 91
 - service level, 91
 - strategy execution, 91
 - technology transformation, 91
 - Component-based development (CBD)
 - methods, 103
 - Computer-aided manufacturing (CAM), 99–100
 - Computer science (CS), 2, 7, 9, 16, 19, 48, 53–54, 68, 70, 84, 151–152, 192, 256–257
 - Computing, technology components
 - communication networks, 88
 - computing platform, 88
 - software, 88
 - Computing and information technology (CIT), 9
 - Conceptual blockbusting theory
 - blocks classified by Jim Adams, 148
 - Concrete IT applications
 - general solution concepts, 49
 - meta artifacts
 - IT product, 49
 - system development process, 49
 - specific solution concepts, 49
 - theses 5, 49–50
 - Consumer phase, 80, 243
 - Contingency theory, 34, 201–202
 - Control group, 117
 - Controlled-flexible development methods, 104–105
 - CORBA, 92, 102–103
 - COTS, *see* Commercial off-the-shelf (COTS)
 - CPM, *see* Cyclical process model (CPM)
 - Creativity
 - De Bono/Couger's view, 145–146
 - key components, 146
 - Nickerson steps, 146
 - Critical realism, 216
 - Critical realist approach
 - developing knowledge, 224
 - identify problem situations/desired outcomes, 222
 - kernel theories/previous research, 221f, 222
 - major activities, 221
 - propose/refine design theory, 222–223
 - test design theory, 223
 - See also* Design theory
 - knowledge produced
 - community demands, 217
 - constituencies, 217
 - IS professionals/managers, 217
 - type of knowledge produced
 - Benbasat and Zmud's view, 218
 - PIMCO configurations, 220–221
 - solution technology invention, 218
 - CS, *see* Computer science (CS)
 - Cyclical process model (CPM), 99, 183t, 184, 186–189
- D**
- Data Base Task Group (DBTG)
 - data schemas, 96
 - conceptual schema, 96
 - external subschema, 96
 - physical schema, 96
 - Data-centric paradigm, 90, 102
 - Data-flow architecture, 89
 - Data management, current trends, 97
 - major challenges, 97
 - Data-oriented methods, 102–103
 - Data-oriented system, 102
 - Data quality metrics description
 - dimensions, 132, 133t
 - research landscape, 132f
 - DBTG, *see* Data Base Task Group (DBTG)
 - DCOM, 92, 103
 - Department of Defense developers, 93
 - Descriptive methods in evaluation, 119–120
 - Design, characteristics, 2
 - Design/creativity
 - in age of virtual worlds, 152–153
 - examples, 152
 - growth rate, 152
 - various fields, 153

- along with IT, 150–152
 - design principles, 150–151
 - conceptual blockbusting theory, 148–150
 - definition, 145–147
 - designing virtual worlds, 153–154
 - paradigm shift by Bricken, 154
 - experiential learning, 150
 - group creativity, 147–148
- Design gestalt, 203
- Design method, 12, 16, 38, 40, 50, 52, 102, 182, 188, 212
- Design of emerging digital services, taxonomy
 - fundamental design dimensions
 - malleability, 244–245
 - pricing/funding, 245–247
 - service delivery, 242
 - service maturity, 243–244
 - fundamental service provider objectives
 - business objective, 247–248
 - interaction objectives, 248–249
 - technological objectives, 248–249
 - future research considerations, 252
 - research objectives, 238–240
 - Craigslist.com, 240
 - digital service providers, sample list, 239t
 - service vs digital service, 237–238
 - taxonomy, 240
 - evaluation, 250–252
 - grounding, 240–241
 - summary, 249–250
- Design process knowledge
 - technical norms, 51, 52t
 - types of, 51, 52t
- Design product knowledge, 48, 51–52, 52t
 - aspects of, 51
- Design research, 187
 - outputs by March/Smith, 181
- Design research criteria to action research exemplar
 - communication of research
 - applications, 189t
 - design as an artifact
 - definition, 187
 - primary contributions, 187
 - design as a search process, 188–189
 - design evaluation, 187–188
 - problem relevance, 187
 - aim, 187
 - research initiated at Scandinavian financial institution, 187
 - research contributions, 188
 - research rigor, 188
- Design science research (DSR)
 - activities
 - build process, 197
 - evaluate process, 197
 - in context, 5–6
 - definition, 5
 - design perspectives, 1–2
 - design science, reasoning, 3–5
 - stokes matrix quadrant, 3f
 - difference between routine design practice and, 7–8
 - research, definition, 2–3
 - phenomenon, 2
 - understanding, 2
 - spectrum of IS DSR, 6–7
 - work scale, discipline balance and scope, 7f
- Design Science Research in Information Systems & Technology (DESIRIST), 13, 257
 - multi-disciplinary attendance and agenda, 13
- Design Science Research Methodology (DSRM), 28–30
- Design theory
 - artifact development vs., 6, 277
 - characteristics, 52–53
 - components of, 211–212
 - concept of, 33–42
 - e-learning, role of, 226
 - framework, 289–290
 - kernel theory and, 52–53, 222
 - knowledge management systems and, 224
 - M&A, (Accenture's survey 2006), 227–228
 - organizations, impact of, 201
 - philosophical groundings, 214
 - propositions, 222–223, 225, 227–228, 285
 - scholar's perception, 263
 - testing, 221, 223, 225–227
 - user requirements, 185, 222–223
- Design theory #1
 - identify problems/desired outcomes
 - aims, 224
 - goal, 224
 - primary cause, 224
 - propose/refine design theory
 - design proposition #2, 225
 - design proposition #9, 225
 - KMS initiative, purpose, 225
 - qualitative study with three phases, 225
 - study on 38 plants, 225
 - test design theory, 225

- Design theory #1 (*cont.*)
 - review (Kernel) theories/previous research focused on, 224–225
 - KMS management into KMS use, 224
 - Design theory #2
 - design proposition #5, 227
 - design proposition #7, 227
 - identify problems/desired outcomes
 - e-learning, comparison, 226
 - propose/refine design theory
 - examples, 227
 - participation, 227
 - review (Kernel) theories/previous research
 - cognitive model, 226
 - test design theory
 - empirical gamma test, 227
 - Design theory #3
 - identify problems/desired outcomes
 - survey by Accenture, 227
 - review (Kernel) theories/previous research, 228
 - Design trade-offs, 64, 88, 250
 - DESRIST, *see* Design Science Research in Information Systems & Technology (DESRIST)
 - Developing a Discipline of Design/Science/Research* (2001), 6
 - Disciplinary theses
 - applied/practical discipline, 44–45
 - building IT artifacts, design science activity, 47–48
 - prescriptive research, 45–47
 - Discovery process, 197
 - See also* Natural science research, activities
 - Dissemination of design science research
 - commercializing ideas *via* start-ups, 259
 - angel investors, 259
 - entrepreneurs, 259
 - conference/journal papers, academic route, 255–257
 - mantra “publish or perish,” 255
 - rates, 256–257
 - funding to support
 - budget structure, 258
 - governmental/corporate agencies, 258
 - proposal format, 257–258
 - sources, 257
 - Distributed architecture, 90, 98
 - vital advantages, 90
 - Domain theories, 72
 - Dot.com days, 80, 84, 246
 - DSR, *see* Design science research (DSR)
 - DSR cycles, 18–19
 - relevance cycle, 17
 - field testing, results, 17
 - rigor cycle, 17–18
 - knowledge, types, 17–18
 - DSR frameworks
 - action research framework, 27–28
 - diagnostic stage, 27
 - five phases, 28
 - therapeutic stage, 28
 - DSRM process, 27–30
 - communication, 30
 - demonstration, 30
 - design and development, 29–30
 - evaluation, 30
 - objectives for solution, definition, 29
 - problem identification and motivation, 28–29
 - general design cycle (GDC), 26–27
 - empirical method, 27
 - reasoning, 27f
 - natural/artificial worlds, understanding, 23–24
 - perspectives, 24
 - systems development in IS research, 25–26
 - research model, 26f
 - toward theory of complex systems, 24–25
 - DSR in information systems, 9
 - extending the reach, 14–21
 - design as research *vs.* researching design, 15–16
 - DSR checklist, 19, 20t
 - DSR cycles, 16–19, 16f
 - DSR *vs.* professional design, 15
 - publication of DSR, 19–21
 - three design research cycles, questions, 20f
 - MISQ* paper (2004), summary, 10–12
 - complementary nature of two paradigms, 11f
 - DSR guidelines, 12t
 - impact on DSR, 13–14
 - DSRM, *see* Design Science Research Methodology (DSRM)
 - DSR’s role, information system, *see specific theories*
 - Dyads, 82
 - Dynamic system theories, 71–72
- E**
- Early computer interaction, 98
 - blinking lights, 98
 - card readers/punches, 98

- paper tape punches, 98
 - primitive cathode-ray tubes, 98
 - teletype printers, 98
 - toggle switches, 98
 - E-commerce, 13, 81, 114
 - Edisonian experiments, 3–4
 - Efficacy, 12, 29, 41, 131, 138–139, 182, 187, 277, 279–280
 - EFGs/CFGs participant, requirements, 134
 - EKP, *see* Emergent knowledge processes (EKP)
 - Electronic medical records (EMR), 264
 - e-mail value accelerator, 166
 - Emergent controls, software design
 - ongoing feedback, 105
 - scope boundaries, 105
 - Emergent knowledge processes (EKP), 184–185, 278, 285, 289, 291
 - Emergent organizational phenomena, 13, 200
 - Empty world hypothesis, 25
 - EMR, *see* Electronic medical records (EMR)
 - End-user computing, 98
 - Engineering design, 2, 4, 146
 - Enterprise JavaBeans, 92, 103
 - Enterprise resource planning (ERP), 49, 218–219
 - business systems, 92
 - Enthusiast phase, 80
 - classic example, 80
 - geeks, 80
 - nerds, 80
 - Epistemic fallacy, 214
 - Epistemological theses
 - knowledge area formation, 50
 - meta-artifacts, process knowledge, 51–52
 - ERP, *see* Enterprise resource planning (ERP)
 - Ethical changes in DSR
 - theses 11, 57–58
 - Ethical theses
 - application of kernel theories, 55–57
 - explicit values, 58
 - IS as design science, 57–58
 - Ethnography, 81–82
 - Evaluation
 - basic structure, 112–113
 - studies of, 112
 - classifications of
 - objectivist, 113
 - subjectivist, 113
 - definition, 110
 - errors in, 118–119
 - mistake avoidance in, 115–116
 - objectives of, 110–111
 - practical, 111
 - promotional, 110–111
 - scholarly, 111
 - objectivist comparative study, example, 116–118
 - performance techniques, 113–115
 - classification, 113–115
 - testing, 113
 - socio-technical aspects, example, 116–118
 - stakeholders perception of, 111–112
 - differing perspectives, 111
 - Evaluation, taxonomy
 - itunes.com, 251–252t
 - myspace.com, comparison, 251t
 - salesforce.com, comparison, 250–251t
 - Evaluation technique selection, 114–116, 119, 123
 - Event-driven architecture, 91
 - Expected uptake, future DSR
 - collaboration/web 2.0/social technologies
 - functions, 267
 - teams at various technologies, 267
 - tools, 267
 - green computing
 - new technologies created/improved, 266–267
 - objectives/goals, 266
 - green technology/green IT
 - carbon footprint, 264
 - renewable energy, 265
 - health care/IT
 - discrepancy between expenditures/health status, 263
 - functions, 264
 - multiple stakeholders, 264
 - Experiential learning
 - experiential learning model by David Kolb/Roger Fry, 150
 - principles, 150
 - Experimental methods in evaluation, 119
 - Explicit value, DSR
 - thesis 12, 58
 - Extensible Markup Language (XML), 100
 - Extreme programming (XP), 104–105
- F**
- Facebook.com, 246
 - Filter architecture, 89
 - Flexibility, 148
 - Focus group, 122
 - Focus group participants
 - sample, 136f

- Focus groups in DSR, use of
 - design research, adapting focus groups
 - data, analyze/interpret, 129–130
 - development/pre-test, 128–129
 - focus group conduct, 129
 - identify moderator, 128
 - identify sample frame, 126
 - participant recruitment, 127–128
 - participants number, 127
 - process, 131f
 - report results, 130
 - research question/problem, formulation, 124–126
 - examples
 - data, analyze/interpret, 137–138
 - data quality metrics description, 131–133
 - design research questions, 133
 - focus groups conduct, 135–137
 - identify sample frame, 133–134
 - questioning route, development, 134
 - recruit participants, 134–135
 - research context, 131
 - goals, 124–126, 125f–126f
 - identify moderator, 134
 - limitations, 139
 - research focus groups, 122–124
 - Focus group technique, 122–123
 - Formal communication, 227
 - Formal development methods, 103
 - FORmula TRANslation (FORTRAN), 93–94, 102
 - FORTRAN, *see* FORmula TRANslation (FORTRAN)
 - Fundamental design dimensions
 - malleability
 - advantages, 244
 - barriers, 245
 - dimensions, 244–245t
 - levels of, 245
 - overall concept, 244
 - pricing/funding
 - generic approaches by Rajala et al., 246
 - network effect, 246
 - operational funds, source, 246
 - revenue logic, 245
 - user value, 245
 - ways to monetize, 246–247
 - service delivery
 - examples, 242
 - requirements, 242t
 - service maturity
 - customer service delivery system, 243
 - embedded systems, 244
 - four stages, 243t
 - fully automated systems, 244
 - technical systems requirement, 243
 - Fundamental service provider objectives
 - business objective
 - functions, 247
 - key questions, 248
 - interaction objectives
 - modular programming techniques, 249
 - Weightman suggestion, 249
 - technological objectives
 - open-source Mysql database, 248
 - Future design science research
 - design as research method, IS scholars, 262
 - build–evaluate cycles, 262
 - technical grounding, 262
 - expected uptake, 264–265
 - collaboration/web 2.0/social technologies, 267
 - green computing, 266–267
 - green technology/green IT, 264–265
 - growing number of scholars research design, 262
 - case study approach, 262
 - steady number of scholars study design theory, 263
 - issues, 263
- G**
- Gamma test, 223, 225–228
 - GDC, *see* General design cycle (GDC)
 - GDSS, *see* Group Decision Support Systems (GDSS)
 - General design cycle (GDC), *see* DSR frameworks
 - General Electric, 96
 - General systems theory, 89
 - Global database schema, 96
 - Global warming, 264
 - Google, 82, 84–86, 250
 - Google search engine, 80, 82, 84–86, 250
 - philosophies, 85
 - Go-To Statement Considered Harmful* (1968), 94
 - Grand strategy, 242
 - Graphic user interface (GUI), 82, 95, 99
 - Greenhouse gases, measurement, 264–267
 - Group creativity
 - face-to-face/electronic groups, comparison, 147
 - proponents, 147
 - Group Decision Support Systems (GDSS), 10

GUI, *see* Graphic user interface (GUI)

Gyro Gearloose style, 55

H

The Handbook of Creativity, Robert Sternberg, 146

HCI, *see* Human–computer Interaction (HCI)

Herman Miller, 82

Hierarchical data model, 96

Hollerith card, 98

HTML, 100, 267

Human–computer Interaction (HCI), 50, 82, 98–100, 154

Human–computer partnership, 76

Human–software interface, 65

Hypothetico-deductive method, 197

I

IBM, 93, 95, 97

magnetic drum, 95

magnetic tape, 95

punched card, 95

research laboratory, 97

IDEO, 82, 202

Incentive structures, 14, 199

Incremental development, 101

emphasis, 101

Indexed sequential access mechanism (ISAM), 96

Information acquisition, 131

Information engineering, 102

Information supply chains (ISC), 131–132

Information system, DSR guidelines

application of kernel theories, 55–57

application typology (IT) artifacts

functions, 48–49

theses 4, 48–49

applied/practical discipline, 44–45

conceptual level, 46f

descriptive level, 46f

as potential role, 57–58

prescriptive level, 46f

theses 1, 44–45

building IT artifacts, design science activity, 47–48

constructive research methods, 53–55

descriptive discipline

theses 3, 47–48

design theory, characteristics, 52–53

explicit values, 58

IS as a design science, 57–58

knowledge area formation, 50

meta-artifacts, concrete IT applications, 49–50

meta-artifacts, process knowledge, 51–52

prescriptive discipline

theses 2, 45–47

prescriptive research, 45–47

sound kernel theory vs. design theory, 52–53

sound ontology, 48–49

Information Systems Design Theory (ISDT), 37–40, 211

components, 38

levels of usage, 39–40

structural components, 39t

towards anatomy of, 41, 42f

artifact mutability, 41

constructs, 41

form and function, principles, 41

justificatory knowledge, 41

purpose and scope, 41

testable proposition, 41

See also On design theory, design theory possibility

Information systems (IS), design science research

alternative philosophies/positivism in IS/DSR field, 211–214

components, 211t

critical aims, 213

focus on IT artifact, 213–214

guidelines, 212

IS community, debate, 213

philosophies, 213–214

critical realism

domains outlined by Bhaskar, 215–216

idealism/realism, contradiction, 215–216

ontological assumptions, 216t

Roy Bhaskar concept, 215

critical realist approach

design theory, 224–229

developing knowledge, 221–223

examples, 224

knowledge produced, 217–218

type of knowledge produced, 218–221

Information technology (IT), 6

artifacts, *see* IT artifacts

definition, 5–7

use of, 10

Information volatility metric, 133

efficacy, 139t

utility, 138t

Integrated research process, envisioning

four-stage model, 191

synthesized research approach, 192f

Intel, 82, 266
 Internet-based retail, 81
 Internet protocol, 237
 Intuitionist-pluralist method, 113
 See also Evaluation, classifications of
 iPod, 7
 ISAM, *see* Indexed sequential access
 mechanism (ISAM)
 ISC, *see* Information supply chains (ISC)
 IS core, 213, 218
 IS design theory, *see* Kernel theory
 ISDT, *see* Information Systems Design Theory
 (ISDT)
 IS-specific theory, 52–53
 IS theory, classification, 37–38
 ISO-9000, 102
 IS professional, role, 217
 IS World web site, 214
 IT, *see* Information technology (IT)
 IT artifacts, 9, 44–45, 47–51, 54–55, 57,
 59–60, 75–76, 110, 116, 196,
 209–210, 212–213, 218–219
 definition, 6
 design theories *vs.*, 211–213
 DSR activity
 theses 3, 48
 theses 4, 48–49
 theses 6, 50
 theses 7, 51
 theses 9, 54–55
 theses 10, 57
 theses 11, 58
 emancipatory and classical issues, 218
 IS core and, 219
 in IS research, 9, 44, 261
 practical implications, 196
 prescriptive research and, 47
 pure, 218
 qualities, 75
 science of design, 60, 76
 socio-technical evolution of, 116
 software and hardware, 110
 technical norms, 50
 weak link, 50
 See also Meta artifact
 IT innovations, 11, 56
 iTunes.com, 236

J
 Java (programming language), 95
*Journal of the American Medical
 Association*, 263
 Justification, 197
 See also Natural science research, activities

K
 Kernel theory
 definition, 52–53
 design process, 38
 design product, 38
 IS specific theory, 53
 theses 8, 52–53
 Knowledge flow forms, Nonaka, 159
 barriers examples, 160
 Knowledge Management Systems (KMS),
 design language
 artifact construction, 162
 iGrafx simulation software, 162
 barriers, 164–165
 concept, 159–162
 conceptual model, 159f
 graphical constructs, 161t
 value accelerators, examples, 160
 evaluation methodology/SME model
 instantiation, comparisons, 167–169
 basic model, 168f
 study on evaluation instantiations,
 167–168, 170f–171f
 good packets, receiver, 167
 knowledge packet generator, 162–164
 problem statement, 159
 results, 169–174
 contributions, 174–175
 summary, 175t
 value accelerators, 165–167
 Knowledge packet generator, 162
 explicit packets, distribution, 164t
 exponential probability function, 162
 information packet, 163
 usefulness delay, 165t
 β value, 163t

L
 Leading digital service, 240, 244, 250
 Learning Lab Denmark (LLD), 204
 LLD, *see* Learning Lab Denmark (LLD)
 Logical-positivist approach, 113
 See also Evaluation, classifications of
 Lotus 1-2-3, 99, 238

M
 Make and sell strategy, 91
 Management disciplines, design science
 design concepts, 198–200
 definition, 198
 organizational design, 199
 as wicked problem, 198
 research in organizational studies, 200–204
 empirical results on study, 201

Management Information Systems Quarterly (MISQ), 10–16, 19

Mapping, 88

Mario Bunge's view on critical realism, 214

Measurement techniques, 114

Mental (small-scale) model, 28

Meta artifacts, IT product

components, 49

construction, 53–55

rigorous research, 55

scientific evaluation of, 55

specific research method, 55

theses 9, 53–55

in design theories, 51

IS design science theses, 49, 51, 53–56, 59

in rigor cycle, 18

implications in DSR, 51

knowledge base and, 51

system development

components, 49

theses 7, 51–52

Meta-design class, 38, 40, 49, 52, 212

Meta-requirement class, 29, 38–40, 42, 49, 52, 212

Methodological theses

constructive research methods, 53–55

sound Kernel theory vs. design theory, 52–53

Metrics, 114

Microsoft, 82, 99–100, 240, 246

Microsoft Excel, 99

Microsoft's PowerPoint, 99, 134–135

Mini-focus groups, 82

MISQ, see *Management Information Systems Quarterly (MISQ)*

MISQ paper, primary goal, 12

MIS Quarterly, 200

MIT, 98

Moderator, role, 135, 139

Mosaic (web browser), 84, 100

Multi-methodological approach, 25

MVS operating system, 99

Myspace.com, 235–236, 239, 246, 251

N

National Science Foundation (NSF), 64

NATO software engineering conference (1968), 93

Natural science research

activities, 197

method, 10, 13

The Nature of Theory in Information Systems, 37

Near-decomposability property, 25

Neil's Bohr quadrant, 3

Network effect, 246

Network data model, 96

Notion of complexity, 23

NSF, see National Science Foundation (NSF)

NSF federal agency, 3–4, 6, 64, 150, 258

O

Objectivist comparative study, in evaluation

before-after study, 117

dependent variable, 116

independent variable, 116

Object orientation, software design

concepts, 103

emphasis on system architectures, 103

encapsulation and class libraries, 103

risks, 103

Observational case studies, in evaluation, 119

On design theory, 33

design theory possibility, 37–41

Hooker's view, 40–41

ISDT, 38–40, 40f

theory in IS, different types, 38t

towards anatomy of IS design theory, 41

theory, definition, 33–34

theory building cycle, 34–36

anomaly-improving descriptive theory, 36

classification, 35

defining relationships, 35–36

observation, 34–35

stages of, 34f

theory types in IS, taxonomy, 37

transition to normative theory, 36–37

One-on-one interviews, 82

Online discussion group technique, 82

Ontological, theses

meta-artifacts, concrete IT applications, 49–50

sound ontology, 48–49

Oracle, 92

Organizational Science, 200

Organization Studies, 200

P

PageRank technique (US Patent 6285999), 85

Paradigmatic/multi-paradigmatic communities, 7

Party groups, 82

Pascal, 94

Pasteur's quadrant, 4

Patterns, 151

Pediatric Intensive Care Unit, 202

- Performance evaluation, 114
 - Performance evaluation mistakes
 - analysis without understanding problem, 115
 - incorrect performance metrics, 115
 - no goals, 115
 - unsystematic approach, 115
 - wrong evaluation techniques, 115–116
 - Phenomenon, 2
 - PIMCO configuration, 220–221
 - Pipe and filter architecture, 89
 - Pipe architecture, 89
 - Possibility of Naturalism*, Roy Bhaskar, 215
 - Pre-theory, 40
 - Primary footprint, 265
 - Procedure-oriented methods, 102
 - Professional phase, 80
 - Project dimensions, software design, 104
 - Propose/refine design theory, 222–223, 227
 - Pure science vs. applied science, 51
 - Push-pull concept, 160
- Q**
- Quality metrics, 130–133, 135, 137–138, 140
 - Quinn's indeterminacy of translation thesis, 41
- R**
- Rapid prototyping approach, 104
 - Rational unified process (RUP), 104–105
 - RCA, *see* Researcher–client agreement (RCA)
 - R&D, *see* Research and development (R&D)
 - The Reflective Practitioner* (1983), 83
 - Regression analysis technique, 35
 - Relational algebra method, 97
 - Relational calculus method, 97
 - Relational database, 68, 97
 - commercialization, impact of, 97
 - Reminder group, 117
 - Repository architecture, 90
 - Research, characteristics, 3
 - Research and development (R&D), 64–65, 68, 73, 93–94, 96, 98–100
 - Researcher–client agreement (RCA), 183–184, 186
 - Research focus groups, definition, 122
 - Research methodology, 2–3, 26, 28–30
 - Resource allocation process, 35, 90, 132
 - Return on investment (ROI), 266
 - Rigorous testing, 200
 - The Rise of the Creative Class*, Richard Florida, 150
 - ROI, *see* Return on investment (ROI)
 - Rolling interview guide, EFG, 128, 134
 - Roy Bhaskar concept, 215
 - RUP, *see* Rational unified process (RUP)
 - RUP approach, 105
- S**
- Safety-critical system, 103–104
 - Salesforce.com, 235–236, 239, 243, 250–251
 - SAP, 92
 - Sample size indicator, 133t
 - Secondary footprint, 265
 - Sciences of the Artificial* (1960 & 1996), 6, 10–11, 23–24, 204, 211
 - Scientific design vs design science, 59–60
 - Scientific method/theory, 3–4, 13, 182, 197, 236, 240
 - organizational phenomena, exist/emerge, 13
 - Scrum, 104
 - SE, *see* Software engineering (SE)
 - SEI, *see* Software Engineering Institute (SEI)
 - Sense and respond strategy, 91
 - Service-oriented architectures (SOA), 92
 - Service vs digital service
 - Amazon.com, 237
 - definition, 237
 - digital/ownership rights, comparison, 238
 - normal/digital services, differences, 238
 - service providers, 238
 - tangibility, 237
 - Short quotes, 130
 - Simula, 94
 - Simulation techniques, 114
 - Simultaneous nonrandomized controls, 117
 - SIS, *see* Software-intensive systems (SIS)
 - Small- to mid-sized enterprises (SMEs), 158
 - SMEs, *see* Small- to mid-sized enterprises (SMEs)
 - SOA, *see* Service-oriented architectures (SOA)
 - Social loafing, 127
 - Social Origins of Educational Systems*, Margaret Archer, 215
 - Socio-economic theories, 72
 - Socio-psychological phenomenon, 40
 - Software artifacts, building, 2
 - Software design, past and present, 2, 79, 87
 - algorithmic design, 92–95
 - early program, 93
 - recent algorithm paradigms, 94
 - structured program, 93–94
 - structures, 94
 - widely used programming languages, 94–95

- data design, 95–97
 - computerized file management, 95–96
 - online data processing, 96
 - punched card data management, 95
- framework, 87–88, 88f
- HCI design, 98–100
 - current trends in, 99–100
 - vital directions, 100
 - WIMP interface, 99
- software architecture, 88–92
 - component-based, 91–92
 - distributed/client–server, 90–91
 - mainframe, 89
 - manual business process, 88
 - online/real-time, 89–90
 - service-oriented, 92
- software development processes/methods, 100–105
 - agile development, 104
 - component-based development (CBD), 103
 - controlled-flexible development, 104–105
 - early development, 102
 - focus, 102
 - formal development, 103
 - object-oriented (OO), 102–103
- See also* Data management, current
 - trends; Early computer interaction;
 - Relational database; Text-based command interface
- Software development process / method, 88f, 100–101
- Software engineering community, 123
- Software (engineering) design, 2, 4, 64, 70–72, 74, 83, 87–105, 145–146, 238
- Software Engineering Institute (SEI), 101
- Software engineering (SE), 2, 4, 7, 16, 48, 53–54, 70, 88, 93, 101, 123, 256–257
- Software-intensive systems (SIS), 65
 - primary activities by skilled team, 72
 - adaptation, 74
 - architecture, 73
 - imagination, 73
 - operations, 73
 - realization, 73
 - principles, categories, 65
 - adaptability, 69
 - computational, 69
 - creative, 69
 - decidability, 69
 - economic, 69
 - ethical, 69
 - scalability, 69
- Software-intensive systems (SIS), science of design, 63
 - concept, 65–66
 - layers, 65f
 - trends, 66f
 - design challenges, science, 63–64
 - objectives, 64
 - principal reasons, 64
 - design science, principles, 66–68
 - concepts in software layer, 67f
 - current CSIS development principles, 68t
 - SIS, physical/software/social networks, 67f
 - engineering activities, 72–74
 - primary activities, 73–74
 - intellectual drivers for transformative research, 75–76
 - complexity, 75
 - composition, 75
 - control, 76
 - principles, categories, 68–69
 - proposed research vision, 69–70
 - science of design in SIS research vision, 70f
 - research project framework, 74
 - examples, 74
 - scientific theories, 70–72
 - domain theories, 72
 - dynamic system theories, 71–72
 - research on science of design in SIS, 71f
 - socio-economic theories, 72
 - software design theories, 70–71
- Software–platform interface, 65
- Software system self-awareness, 76
- Sophisticated indexing technique, 96
- SOPs, *see* Standard operating procedures (SOPs)
- Spaghetti code program, 94
- Spiral model, 101
- SQL, *see* Structured query language (SQL)
- Standard operating procedures (SOPs), 89
- Stanford, 98
- Strategic alignment model, 91
- Structured query language (SQL), 97
- Sumerian stone tablet, 95
- Sun Microsystems, 95, 153
- Sun microsystems in Second Life, 153
- Super groups, 82
- System, 65

- Systematic evaluation process, 116
 System descriptions, 25
 Systems development framework, 26
- T**
- Tasks classified by Schneiderman, 151
- Taxonomy
 classical approach, 241–242
 grounding
 Aristotle's *Rhetoric*, 241
 brainstorming differences, 240
 principles from roman architect
 Vitruvius, 241
 summary
 digital service design taxonomy, 249f
- Technology, 5–7, 9–13, 17, 19, 25, 48–51, 54,
 75, 79–80, 82, 84, 87–88, 90–91,
 97, 99, 104, 110–111, 151–152,
 154, 161, 163, 180–182, 184, 188,
 204–205, 210, 213, 226, 236–237,
 248, 258, 262–264, 266, 269
- Technology and domain of human concerns, 79
 designing for consumers, 80–81
 designing for scale—Google and people,
 84–86
 ethnographic practice in design, 81–83
 chasm crossing, challenge, 82
 imponderabilia of actual life, 81
 qualitative research methods, 82
 situatedness, 81
 as working concept, 81
 reflective stance in design (Schon's view),
 83–84, 84f
 innovative methods, 83
- Technology organization and people
 integration modeler (TOP modeler),
 183–184
- Template analysis, 130
 definition, 137
 partial final coding scheme, 137f
- Testable design product hypotheses, 38, 40,
 52, 212
- Test design theory, 223, 225–229
- Text-based command interface, 98–99
 standard applications, 98–99
 computer-aided design (CAD), 99
 presentation and graphics, 99
 spreadsheets, 99
 text editing and word processing, 98–99
- Theory/theories, *see specific theories*
- Theory building process, DSR, 34
 conceptual information modeling, 56–57
 major sources, 56
 object-oriented systems development, 57
 personal productivity tools, examples, 57
 theses 10, 55–57
- TOP modeler, *see* Technology organization
 and people integration modeler
 (TOP modeler)
- Traditional focus group, 82, 127, 140
- Triads, 82
- U**
- UC Davis Virtual Hallucinations facility in
 Second Life, 153
- UML, *see* Unified modeling language (UML)
- Unified modeling language (UML), 103
- UNIVAC I computer system, 95
- UNIX operating system, 89, 94, 99
- URL, 100
- USA, 80–81, 86, 227, 229, 239, 251, 259
- Utility, 138
 theory, 34
- V**
- Value accelerators
 brown bag meetings, 166
 e-mail, 166
 knowledge repositories, 166
- VAX VMS operating system, 99
- VB, *see* Visual Basic (VB)
- Vignettes, 134–135, 139
 examples, 135t
- Virtual sequential access mechanism (VSAM),
 96
- VisiCalc, 99
- Vista operating system, 81
- Visual Basic (VB), 95
- VSAM, *see* Virtual sequential access
 mechanism (VSAM)
- W**
- Warnier–Orr method, 102
- Waterfall process, 101
 DOD-STD-2167A, 101
- Wicked problems, 11, 87, 198, 202, 262
 characteristics, 11
- WIMP features
 Apple Lisa, 99
 Apple Macintosh, 99
 Xerox Star, 99
- WITS, *see* Workshop on Information
 Technology and Systems (WITS)
- Workloads, 114
- Workshop on Information Technology and
 Systems (WITS), 9–10, 257
- World Wide Web (WWW), 56, 92, 97, 99–100
- X**
- Xerox PARC, Star project, 82, 98–99
- XML, *see* Extensible Markup Language
 (XML)
- XP, *see* Extreme programming (XP)