Journal of Management Information Systems

ISSN: 0742-1222 (Print) 1557-928X (Online) Journal homepage: https://www.tandfonline.com/loi/mmis20

A Design Science Research Methodology for Information Systems Research

Ken Peffers , Tuure Tuunanen , Marcus A. Rothenberger & Samir Chatterjee

To cite this article: Ken Peffers , Tuure Tuunanen , Marcus A. Rothenberger & Samir Chatterjee (2007) A Design Science Research Methodology for Information Systems Research, Journal of Management Information Systems, 24:3, 45-77, DOI: 10.2753/MIS0742-1222240302

To link to this article: https://doi.org/10.2753/MIS0742-1222240302

Published online: 08 Dec 2014.

A Design Science Research Methodology for Information Systems Research

KEN PEFFERS, TUURE TUUNANEN, MARCUS A. ROTHENBERGER, AND SAMIR CHATTERJEE

KEN PEFFERS is an Associate Professor and Chair of the Management Information Systems Department at the University of Nevada, Las Vegas. He earned his Ph.D. in Management Information Systems from Purdue University. His current research focuses on making the right IS investments for the firm, on IS planning, and on requirements determination for new information systems. His research articles have been published in journals such as *Communications of the ACM, Journal of Management Information Systems, Information Systems Research, IEEE Transactions on Engineering Management, Organization Science, Journal of Information Technology Theory and Application,* and *Information & Management.*

TUURE TUUNANEN is a Senior Lecturer in the Department of Information Systems and Operations Management at the University of Auckland. He holds a D.Sc. (Econ) from the Helsinki School of Economics. His current research interests lie in the areas of IS development methods and processes, requirements engineering, risk management, and convergence of IS and marketing disciplines, specifically in design of interactive consumer services and products. His research has been published in *Information & Management, Journal of Database Management, Journal of Information Technology Theory and Application, Journal of Management Information Systems,* and *Technology Analysis and Strategic Management.* In addition, his work has appeared in a number of conference proceedings within his research interest areas such as European Conference on Information Systems, Design Science Research in Information Systems and Technology, Hawaii International Conference on System Sciences, Mobility Roundtable and Requirements Engineering Conference. Dr. Tuunanen is a member of the Association of Computing Machinery, the Association of Information Systems, and the Institute of Electrical and Electronics Engineers.

MARCUS A. ROTHENBERGER is an Associate Professor of Management Information Systems at the University of Nevada, Las Vegas. He holds a Ph.D. in Information Systems from Arizona State University. Dr. Rothenberger's work includes theory testing, theory development, and design science research in the areas of software process improvement, software reusability, performance measurement, and the adoption of enterprise resource planning systems. His work has appeared in major academic journals, such as *Decision Sciences, IEEE Transactions on Software Engineering, Communications of the ACM,* and *Information & Management.* Dr. Rothenberger is regularly involved in major academic conferences, including the *International Conference on Information Systems* and the *Americas Conference on Information Systems.* He is a member of the Association for Information Systems and the Decision Sciences Institute.

Journal of Management Information Systems / Winter 2007–8, Vol. 24, No. 3, pp. 45–77.

0742–1222 / 2008 $9.50 + 0.00.
DOI 10.2753/MIS0742-1222240302

SAMIR CHATTERJEE is a Professor in the School of Information Systems & Technology and Founding Director of the Network Convergence Laboratory at Claremont Graduate University, California. He holds a Ph.D. in Computer Science from the University of Central Florida. He is widely recognized as an expert in the areas of next-generation networking, voice- and video-over Internet protocol, and network security. His current research is on designing IT solutions to problems in health care and collaboration. He has published over 75 articles in respected scholarly journals and refereed conferences including *IEEE Network, IEEE Journal on Selected Areas in Communications, Communications of the ACM, Computer Networks, International Journal of Healthcare Technology & Management, Telemedicine & e-Health Journal, Information Systems Frontiers, Computer Communication, IEEE IT Professional, ACM Computer Communication Review, Communications of AIS,* and *Journal of Internet Technology,* among others. He is the principal investigator on several National Science Foundation (NSF) grants and has received funding from numerous private corporations for his research. He is on the editorial board of *International Journal of Business Data Communications & Networking.* He is the founding Program Chair for the First International Design Science Conference (DESRIST 2006) and General Chair for DESRIST 2008. He has been an entrepreneur and successfully cofounded a startup company, VoiceCore Technologies, Inc., in 2000.

ABSTRACT: The paper motivates, presents, demonstrates in use, and evaluates a methodology for conducting design science (DS) research in information systems (IS). DS is of importance in a discipline oriented to the creation of successful artifacts. Several researchers have pioneered DS research in IS, yet over the past 15 years, little DS research has been done within the discipline. The lack of a methodology to serve as a commonly accepted framework for DS research and of a template for its presentation may have contributed to its slow adoption. The design science research methodology (DSRM) presented here incorporates principles, practices, and procedures required to carry out such research and meets three objectives: it is consistent with prior literature, it provides a nominal process model for doing DS research, and it provides a mental model for presenting and evaluating DS research in IS. The DS process includes six steps: problem identification and motivation, definition of the objectives for a solution, design and development, demonstration, evaluation, and communication. We demonstrate and evaluate the methodology by presenting four case studies in terms of the DSRM, including cases that present the design of a database to support health assessment methods, a software reuse measure, an Internet video telephony application, and an IS planning method. The designed methodology effectively satisfies the three objectives and has the potential to help aid the acceptance of DS research in the IS discipline.

KEY WORDS AND PHRASES: case study, design science, design science research, design theory, mental model, methodology, process model.

INFORMATION SYSTEMS (IS) IS AN APPLIED RESEARCH discipline, in the sense that we frequently apply theory from other disciplines, such as economics, computer science, and the social sciences, to solve problems at the intersection of information technology (IT) and organizations. However, the dominant research paradigms that we use to produce and publish research for our most respected research outlets largely continue to be those of traditional descriptive research borrowed from the social and natural sciences. We recently accepted the use of interpretive research paradigms, but the resulting research output is still mostly explanatory and, it could be argued, not often applicable to the solution of problems encountered in research and practice. While design, the act of creating an explicitly applicable solution to a problem, is an accepted research paradigm in other disciplines, such as engineering, it has been employed in just a small minority of research papers published in our best journals to produce artifacts that are applicable to research or practice.

Without a strong component that produces explicitly applicable research solutions, IS research faces the potential of losing influence over research streams for which such applicability is an important value. For example, we wonder whether the preference for theory building and testing research may help to explain why the center of gravity for research in systems analysis and design—arguably, IS research's *raison d'être*—appears to have moved to engineering, dominated by research streams such as requirements engineering and software engineering. Engineering disciplines accept design as a valid and valuable research methodology because the engineering research culture places explicit value on incrementally effective applicable problem solutions. Given the explicitly applied character of IS practice and the implicitly applied character of IS research, as part of the business academe, we should do so as well.

In recent years, several researchers succeeded in bringing design research into the IS research community, successfully making the case for the validity and value of design science (DS) as an IS research paradigm [20, 31, 55] and actually integrating design as a major component of research [33]. In spite of these successful efforts to define DS as a legitimate research paradigm, DS research has been slow to diffuse into the mainstream of IS research in the past 15 years [56] and much of it has been published in engineering journals.

An accepted common framework is necessary for DS research in IS and a mental model [18, 45, 54] or template for readers and reviewers to recognize and evaluate the results of such research. Every researcher trained in the culture of social science research has mental models for empirical and theory building research that allow the researcher to recognize and evaluate such work, and perhaps one for interpretive research as well. Even if all of these mental models are not exactly the same, they provide contexts in which researchers can understand and evaluate the work of others. For example, if a researcher reviewed an empirical paper that failed to describe how the data were gathered, he or she would probably always regard that as an omission that required notice and correction. Because DS research is not part of the dominant IS research culture, no such commonly understood mental model exists. Without one, it may be difficult for researchers to evaluate it or even to distinguish it from practice activities, such as consulting.

A number of researchers, both in and outside of the IS discipline, have sought to provide some guidance to define DS research [20]. Work in engineering [2, 14, 16, 38], computer science [37, 46], and IS [1, 10, 20, 31, 33, 40, 55, 56] has sought to collect and disseminate the appropriate reference literature [51]; characterize its purposes;

differentiate it from theory building and testing research, in particular, and from other research paradigms; explicate its essential elements; and claim its legitimacy. However, so far this literature has not explicitly focused on the development of a methodology for carrying out DS research and presenting it.

We propose and develop a design science research methodology (DSRM) for the production and presentation of DS research in IS. This effort contributes to IS research by providing a commonly accepted framework for successfully carrying out DS research and a mental model for its presentation. It may also help with the recognition and legitimization of DS research and its objectives, processes, and outputs, and it should help researchers to present research with reference to a commonly understood framework, rather than justifying the research paradigm on an ad hoc basis with each new paper.

Problem Identification: Completing a DSRM for IS Research

When IS researchers started to develop an interest in DS research in the early 1990s, there already was agreement in prior research about the basic difference between DS and other paradigms, such as theory building and testing, and interpretive research: "Whereas natural sciences and social sciences try to understand reality, design science attempts to create things that serve human purposes" [43, p. 55]. Three papers from the early 1990s [31, 33, 55] introduced DS research to the IS community. Nunamaker et al. [33] advocated the integration of system development into the research process, by proposing a multimethodological approach that would include theory building, systems development, experimentation, and observations. Walls et al. [55] defined IS design theory as a class of research that would stand as an equal with traditional social science–based theory building and testing. March and Smith [31] pointed out that design research could contribute to the applicability of IS research by facilitating its application to better address the kinds of problems faced by IS practitioners.

Once this literature provided a conceptual and paradigmatic basis for DS research, Walls et al. [56] expected its widespread adoption within IS, believing that this would lead to IS research having more impact on practice through close ties between DS research and practical applications. Despite the precedents of these early papers, Walls et al. [56] observed that this rush to publish DS research did not occur and that the DS research paradigm had only occasionally been used explicitly in the past ten years. Given that many papers in reference disciplines, such as engineering and computer science, use DS as a research approach and, in doing so, realize benefits from the practical applicability of research outcomes (e.g., [3, 19, 27, 28, 29]), it would seem reasonable that it could also happen in IS.

Toward a DSRM

Some engineering literature (e.g., [14]) has pointed to a need for a common DSRM. Archer's [2] methodology focuses on one kind of DS research, which resulted in building system instantiations as the research outcome, or "the purposeful seeking of a solution" [32, p. 14] to a problem formulated from those desires [32]. Archer [2] believed that design could be codified, even the creative part of it. Archer's industrial engineering research outcomes reflect his views on research methodology. His work included purpose-oriented designs for hospital beds and for mechanisms that prevented fire doors from being propped open. Through this work, he defined six steps of DS research: programming (to establish project objectives), data collection and analysis, synthesis of the objectives and analysis results, development (to produce better design proposals), prototyping, and documentation (to communicate the results). With these steps, he asserted that designers can approach design problems "systematically," by looking at functional-level problems such as goals, requirements, and so on, and by progressing toward more specific solutions [22].

A methodology is "a system of principles, practices, and procedures applied to a specific branch of knowledge" [13]. Such a methodology might help IS researchers to produce and present high-quality DS research in IS that is accepted as valuable, rigorous, and publishable in IS research outlets. For DS research, a methodology would include three elements: conceptual principles to define what is meant by DS research, practice rules, and a process for carrying out and presenting the research.

Principles: DS Research Defined

With just a decade and a half of history, DS research in IS may still be evolving; however, we now have a reasonably sound idea about what it is. "Design science . . . creates and evaluates IT artifacts intended to solve identified organizational problems" [20, p. 77]. It involves a rigorous process to design artifacts to solve observed problems, to make research contributions, to evaluate the designs, and to communicate the results to appropriate audiences [20]. Such artifacts may include constructs, models, methods, and instantiations [20]. They may also include social innovations [52] or new properties of technical, social, or informational resources [24]; in short, this definition includes any designed object with an embedded solution to an understood research problem.

Practice Rules for DS Research

Hevner et al. [20] provided us with practice rules for conducting DS research in the IS discipline in the form of seven guidelines that describe characteristics of well carried out research. The most important of these is that the research must produce an "artifact created to address a problem" [20, p. 82]. Further, the artifact should be relevant to the solution of a "heretofore unsolved and important business problem" [20, p. 84]. Its "utility, quality, and efficacy" [20, p. 85] must be rigorously evaluated. The research should represent a verifiable contribution and rigor must be applied in both the development of the artifact and its evaluation. The development of the artifact should be a search process that draws from existing theories and knowledge to come up with a solution to a defined problem. Finally, the research must be effectively communicated to appropriate audiences [20].

Procedures: A Process Model and Mental Model for Research Outputs

Prior research has introduced principles that define what DS research is [20] and what goals it should pursue [16, 20], as well as practice rules that provide guidance for conducting [2, 16, 20, 38] and justifying it [1, 33, 55]. Nevertheless, principles and practice rules are only two out of the three characteristics of a methodology [13]. The missing part is a procedure that provides a generally accepted process for carrying it out.

Hitherto, IS researchers have not focused on the development of a consensus process and mental model for DS research, such as that called for in engineering literature [16, 38] and required by the IS research discipline. This lack of a consensus-based DS research process model may help to explain why, despite many citations, the message of DS research has not resulted in more research in IS that makes explicit use of the paradigm [56]. Instead, much of the DS research published by IS researchers has been published in engineering journals, where DS behaviors are more the norm. Some of that published in IS journals has required ad hoc arguments to support its validity [5, 9, 34, 36, 42]. For example, in Peffers and Tuunanen [34], the authors use information theory to justify the *use* of an IS planning method, which, in reality, was a designed method. In Peffers et al. [36], the researchers justify their work as a practical extension of another methodology, rather than making explicit design claims. In Rothenberger and Hershauer [42], the authors describe the development of a software reuse measure in the context of a field study and evaluate the artifact using one project of the field company that is treated as a case study.

Defining Objectives of a Solution: Process and Mental Models Consistent with Prior Research

Our overall objective for the paper is the development of a methodology for DS research in IS. We do this by introducing a DS process model, which, together with prior research on DS, provides DS research with a complete methodology. The design of this conceptual process will seek to meet three objectives: it will (1) provide a nominal process for the conduct of DS research, (2) build upon prior literature about DS in IS and reference disciplines, and (3) provide researchers with a mental model or template for a structure for research outputs.

A Nominal Process

Such a process could accomplish two things for DS research in IS. It would help provide a road map for researchers who want to use design as a research mechanism for IS research. Such a process would not be the only way that DS research could be done, but it would suggest a good way to do it. It could also help researchers by legitimizing such research, just as researchers understand the essential elements of empirical IS research and accept research that is well done using understood and accepted processes.

Building on Prior Research

There is a substantial body of research, both within the IS literature and in reference disciplines, that provides us with a tradition to support such a process. A process for DS research should build on this work while integrating its principles into a comprehensive methodology for conducting DS research. There are two sets of applicable literature. One revolves around issues of actually doing academic design work—that is, *design research.* The second set addresses the meta level of conducting research at a higher level of abstraction—*research about design research.* Below, we discuss the differences and how both contribute to meeting this objective.

The *design research* literature contains a large number of references to processes that are described incidentally to the production of research-based designs. Many of these descriptions are specific to research contexts and to the practical needs of design practitioners. In engineering, for example, there have been a number of design research efforts in which the focus has been on processes targeting the production of artifacts [53]. Evbuonwan et al. [15] mention 14 such process models. Many, such as Cooper's StageGate [11, 12], are clearly intended as design or development methodologies, rather than research methodologies or processes, such as the one we are seeking to develop for DS research in IS. Likewise, in computer science, Maguire's [30] human-centered design cycle addresses the specific problems of requirements engineering methods for different situations and, in IS, Hickey and Davis [21] addressed the issue from a functional view. Iivari et al. [23] considered the differences between IS development methods and methodologies and the needs that arise for method development. Processes described in this literature are of interest, but, because they vary widely and are generally context specific, they cannot necessarily be directly applied to the development of a general process for DS research.

The *research about design research* literature is rich with ideas about how to conduct research. This literature, while not providing process models that can be applied directly to the problem of DS research, provides concepts from which we can infer processes. In IS, Nunamaker et al. [33] provided an abstract model connecting aspects of design research, but leave the actual process for conducting it to the researcher's inference. Walls et al.'s [55, 56] IS design theory provides theory at a high level of abstraction from which we can infer a process. Hevner et al.'s [20] and March and Smith's [31] guidelines for DS research influence methodological choices within the DS research process. In the computer science domain, Preston and Mehandjiev [37] and Takeda et al. [46] proposed a "design cycle" for intelligent design systems.

In engineering, Archer [2] and Eekels and Roozenburg [14] presented design process models that could be incorporated into a consensus process. Adams and Courtney [1] proposed an extension of Nunamaker et al.'s [33] system development research methodology via inclusion of action research or grounded theory approach as ways to conduct research. Cole et al. [10] and Rossi and Sein [40] proposed basic steps to integrate DS research and action research. So far, no complete, generalizable process model exists for DS research in IS; however, if we develop such a process model, it should build upon the strengths of these prior efforts.

A Mental Model

The final objective of a DSRM process is to provide a mental model for the characteristics of research outputs. A mental model is a "small-scale [model] of reality . . . [that] can be constructed from perception, imagination, or the comprehension of discourse. [Mental models] are akin to architects' models or to physicists' diagrams in that their structure is analogous to the structure of the situation that they represent, unlike, say, the structure of logical forms used in formal rule theories" [25]. Outcomes from DS research are clearly expected to differ from those of theory testing or interpretative research. A process model should provide us with some guidance, as reviewers, editors, and consumers, about what to expect from DS research outputs. March and Smith [31] contributed to this expectation with their ideas about research outputs. Hevner et al. [20] further elaborated on this expectation by describing DS research's essential elements. A mental model for the conduct and presentation of DS research will help researchers to conduct it effectively.

Design: Development of the Methodology

Development of the methodology required the design of a DSRM process. To accomplish this, we looked to influential prior research and current thought to determine the appropriate elements, seeking to build upon what researchers said in key prior literature about what DS researchers did or should do. Our aim here was to design a methodology that would serve as a commonly accepted framework for carrying out research based on DS research principles outlined above. Rather than focusing on nuanced differences in views about DS among various researchers, we sought to use a consensus-building approach to produce the design. Consensus building was important to ensure that we based the DSRM on well-accepted elements.

A number of researchers in IS and other disciplines have contributed ideas for process elements. Table 1 presents process elements, stated or implied, from seven representative papers and presentations and our synthesis: the components of the DSRM process. The authors agree substantially on common elements. The result of our synthesis is a process model consisting of six activities in a nominal sequence, which we justify and describe here and graphically in Figure 1.

All seven papers include some component in the initial stages of research to define a research problem. Nunamaker et al. [33] and Walls et al. [55] emphasized theoretical bases, whereas engineering researchers [2, 14] focused more on applied problems. Takeda et al. [46] suggested the need for problem enumeration, whereas Rossi and Sein [40] advocated need identification. Hevner et al. [20] asserted that DS research should address important and relevant problems.

> *Activity 1: Problem identification and motivation.* Define the specific research problem and justify the value of a solution. Because the problem definition will be used to develop an artifact that can effectively provide a solution, it may be useful to atomize the problem conceptually so that the solution can capture its complexity. Justifying the value of a solution accomplishes two things: it moti-

Table 1. Design and DS Process Elements from IS and Other Disciplines and Synthesis Elements for a DSRM in IS

Common design process elements	Archer [2]	Takeda et al. [46]	Eekels and Roozenburg [14]	Nunamaker et al. [33]	Walls et al. [55]	Cole et al. [10] Rossi and Sein [40]	Hevner et al. [20]
Problem identification and motivation	Programming, data collection	Problem enumeration	Analysis	Construct a conceptual framework	Meta-requirements, kernel theories	Identify a need	Important and relevant problems
Objectives of a solution			Requirements				Implicit in "relevance"
Design and development	Analysis, synthesis, development	Suggestion, development	Synthesis, tentative design proposals	Develop a system architecture, analyze and design the system, build the system	Design method, meta design	Build	Iterative search process, artifact
Demonstration			Simulation, conditional prediction	Experiment, observe, and evaluate the system			
Evaluation		Confirmatory evaluation	Evaluation, decision, definite design		Testable design process/product hypotheses	Evaluate	Evaluate
Communication	Communication						Communication

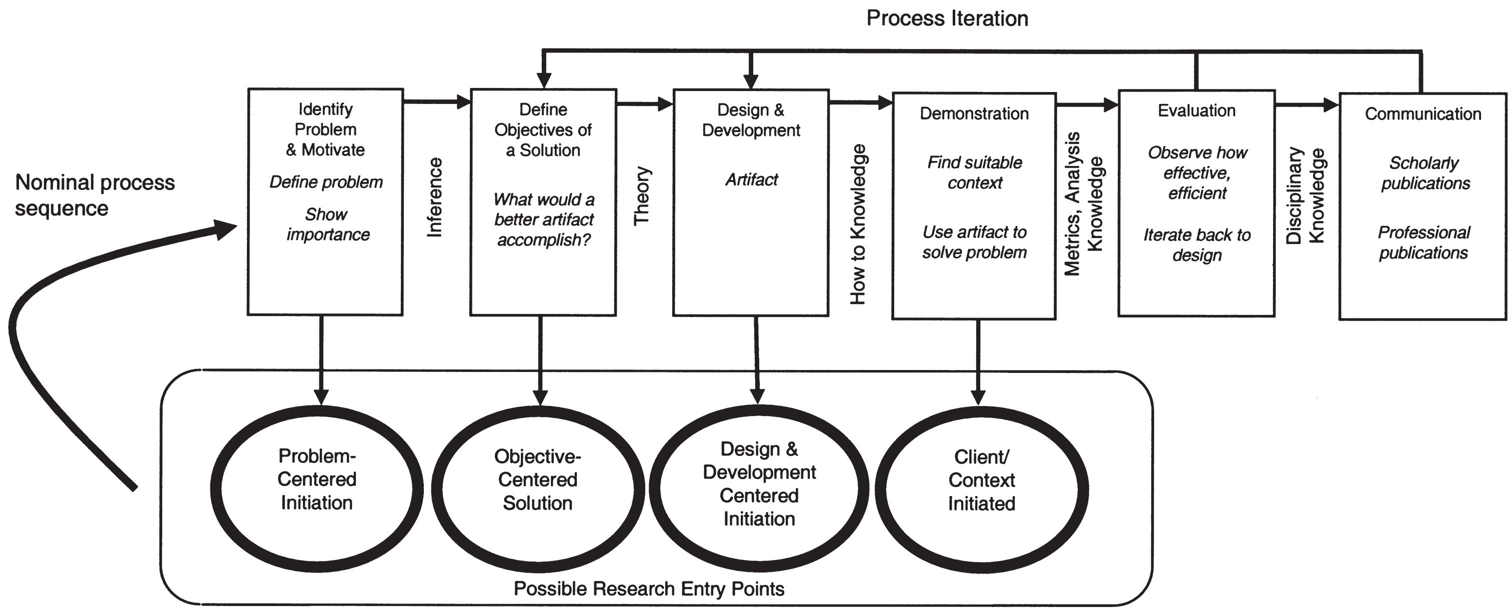

Figure 1. DSRM Process Model

vates the researcher and the audience of the research to pursue the solution and to accept the results and it helps to understand the reasoning associated with the researcher's understanding of the problem. Resources required for this activity include knowledge of the state of the problem and the importance of its solution.

Some of the researchers explicitly incorporate efforts to transform the problem into system objectives, also called metarequirements [55] or requirements [14], whereas for the others, these efforts are implicit as part of programming and data collection [2] or implicit in the search for a relevant and important problem. Identified problems do not necessarily translate directly into objectives for the artifact because the process of design is necessarily one of partial and incremental solutions. Consequently, after the problem is identified, there remains the step of determining the performance objectives for a solution.

Activity 2: Define the objectives for a solution. Infer the objectives of a solution from the problem definition and knowledge of what is possible and feasible. The objectives can be quantitative, such as terms in which a desirable solution would be better than current ones, or qualitative, such as a description of how a new artifact is expected to support solutions to problems not hitherto addressed. The objectives should be inferred rationally from the problem specification. Resources required for this include knowledge of the state of problems and current solutions, if any, and their efficacy.

All of the researchers focus on the core of DS across disciplines—*design and development.* In some of the research (e.g., [14, 33]), the design and development activities are further subdivided into more discrete activities whereas other researchers focus more on the nature of the iterative search process [20].

Activity 3: Design and development. Create the artifact. Such artifacts are potentially constructs, models, methods, or instantiations (each defined broadly) [20] or "new properties of technical, social, and/or informational resources" [24, p. 49]. Conceptually, a design research artifact can be any designed object in which a research contribution is embedded in the design. This activity includes determining the artifact's desired functionality and its architecture and then creating the actual artifact. Resources required for moving from objectives to design and development include knowledge of theory that can be brought to bear in a solution.

Next, the solutions vary from a single act of *demonstration* [55] to prove that the idea works, to a more formal *evaluation* [14, 20, 33, 40, 51] of the developed artifact. Eekels and Roozenburg [14] and Nunamaker et al. [33] included both of these phases.

Activity 4: Demonstration. Demonstrate the use of the artifact to solve one or more instances of the problem. This could involve its use in experimentation, simulation, case study, proof, or other appropriate activity. Resources required for the demonstration include effective knowledge of how to use the artifact to solve the problem.

> *Activity 5: Evaluation.* Observe and measure how well the artifact supports a solution to the problem. This activity involves comparing the objectives of a solution to actual observed results from use of the artifact in the demonstration. It requires knowledge of relevant metrics and analysis techniques. Depending on the nature of the problem venue and the artifact, evaluation could take many forms. It could include items such as a comparison of the artifact's functionality with the solution objectives from activity 2, objective quantitative performance measures such as budgets or items produced, the results of satisfaction surveys, client feedback, or simulations. It could include quantifiable measures of system performance, such as response time or availability. Conceptually, such evaluation could include any appropriate empirical evidence or logical proof. At the end of this activity the researchers can decide whether to iterate back to activity 3 to try to improve the effectiveness of the artifact or to continue on to communication and leave further improvement to subsequent projects. The nature of the research venue may dictate whether such iteration is feasible or not.

Finally, Archer [2] and Hevner et al. [20] proposed the need for *communication* to diffuse the resulting knowledge.

> *Activity 6. Communication.* Communicate the problem and its importance, the artifact, its utility and novelty, the rigor of its design, and its effectiveness to researchers and other relevant audiences such as practicing professionals, when appropriate. In scholarly research publications, researchers might use the structure of this process to structure the paper, just as the nominal structure of an empirical research process (problem definition, literature review, hypothesis development, data collection, analysis, results, discussion, and conclusion) is a common structure for empirical research papers. Communication requires knowledge of the disciplinary culture.

This process is structured in a nominally sequential order; however, there is no expectation that researchers would always proceed in sequential order from activity 1 through activity 6. In reality, they may actually start at almost any step and move outward. A problem-centered approach is the basis of the nominal sequence, starting with activity 1. Researchers might proceed in this sequence if the idea for the research resulted from observation of the problem or from suggested future research in a paper from a prior project. An objective-centered solution, starting with activity 2, could be triggered by an industry or research need that can be addressed by developing an artifact. A design- and development-centered approach would start with activity 3. It would result from the existence of an artifact that has not yet been formally thought through as a solution for the explicit problem domain in which it will be used. Such an artifact might have come from another research domain, it might have already been used to solve a different problem, or it might have appeared as an analogical idea. Finally, a client-/context-initiated solution may be based on observing a practical solution that worked; it starts with activity 4, resulting in a DS solution if researchers work backward to apply rigor to the process retroactively. This could be the by-product of a consulting experience.

Demonstration in Four Case Studies

To demonstrate the use of the DSRM, we apply it retroactively to four already published IS research projects. In the first, researchers design and develop a data warehousing solution to support data gathering and analysis necessary for public health policy [4, 5]. The second explicates the design of a software reuse measure that was used in subsequent case study research [41, 42]. The third reports on the design of an application and middleware for the Internet2 environment that provides telephony and video functionalities [9, 17]. Finally, the fourth depicts the development of a method, critical success chains (CSC) [34, 36], for use in generating a portfolio of new ideas for mobile financial services applications.

In each case, we show how the process of motivating, developing, designing, demonstrating, evaluating, and communicating the artifact is consistent with the DSRM. In none of the cases were the publication outputs explicitly described and presented as using a DS research process, because a designed methodology had not been hitherto available. In the summaries that follow, we used the language of the DSRM to interpret the research processes actually used by the researchers to determine how well the DSRM fits with the research processes used.

Case 1: The CATCH Data Warehouse for Health Status Assessments

The comprehensive assessment for tracking community health (CATCH) methods was published [44] and successfully used in multiple counties in the United States. The methodology requires data to be gathered from multiple sources, including hospitals, health agencies, health-care groups, and surveys. CATCH organizes over 250 health-care indicators into 10 categories that represent a variety of health-care issues. The output of the CATCH methodology is a prioritized listing of community health-care challenges. In this work Berndt and colleagues [4, 5, 7] automated the use of CATCH by developing a data warehouse that implements the methodology. Figure 2 summarizes how the DSRM applies to the steps undertaken as part of this DS research effort [4, 5, 7].

Problem-Centered Approach

The lack of automated support made the data gathering for the CATCH methodology labor intensive and slow; thus, extended trend analyses were cost prohibitive for most communities. The need for a more efficient automated data access for CATCH health assessments triggered the development of the CATCH data warehouse.

Problem Identification and Motivation

The United States has the highest health-care spending of any nation in the world, both as a percentage of gross domestic product (GDP) and per capita. Nevertheless, the United States does not rank among the countries with the healthiest populations. Thus,

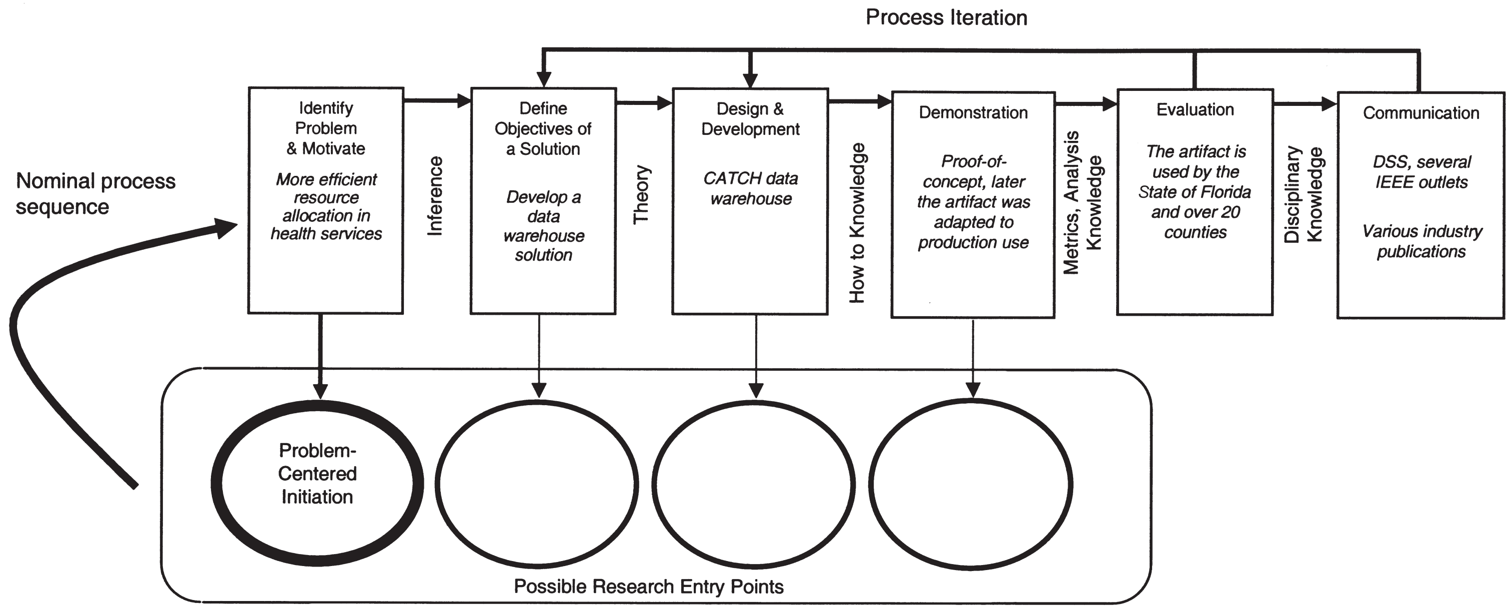

Figure 2. DSRM Process for the CATCH Project

there was a need to assess the country's health status in order to assist communities to develop comprehensive health strategies, leading to better resource allocation for prevention and treatment. The formulation of such strategy had to be based on local health data. The availability and quality of health data was low, which is why health data rarely were the basis for decision making on health policies. Although CATCH was an available assessment method at the time, the labor intensity of nonautomated data gathering limited its adoption.

Objective of the Solution

The objective was to develop a data warehouse solution for the automated support of the CATCH methods that enables users to run cost-effective analyses. The major challenges included the diversity of the data sources, the diversity of target groups for which reports were generated, and the need to conform to the public policy formulation process. The data warehouse was to provide a rich environment that would enable an improvement of research capabilities on critical health-care issues with the long-term goal of centering the role of public health agencies around monitoring and improving the health status of the population using this technology.

Design and Development

The artifact is the data warehouse that supports and automates CATCH. The researchers drew from data warehousing research to develop the CATCH data warehouse with data arranged in a star schema. The design includes three levels of granularity: the report structures, aggregate dimensional structures, and fine-grained and transaction-oriented dimensional structures. Staging and quality assurance methods were established to enable a successful use of the data warehouse and performance issues were addressed. The design and related methods have been and continue to be refined based on emerging performance needs.

Demonstration

After developing proof-of-concept-level prototypes, the artifact was extensively adapted to production use by user organizations. The researchers point to the application of the CATCH data warehouse in multiple counties and provide screenshots of several output screens in their articles. In related research, it was also demonstrated that the CATCH data warehouse could be used to conduct bioterrorism surveillance: a similar data warehouse approach was used in a demonstration surveillance system in Florida.

Evaluation

The original CATCH methods have been used and refined for more than 10 years in more than 20 U.S. counties. The researchers implemented the CATCH data warehouse as a fully functional version in Florida's Miami-Dade County. The verification of the accuracy of the automated generation of the report through a comprehensive manual check identified only minor problems in use. The data warehouse was found to be flexible and effective in this field application.

Communication

Manuscripts relating to the CATCH data warehouse have been published in academic journals, academic conference proceedings, and professional outlets. The development of the health-care data warehouse was presented in *Decision Support Systems* [5] and *Upgrade,* a professional online magazine [4]. The challenges of quality assurance in the CATCH data warehouse were discussed in *IEEE Computer* [7]. Further, the use of data warehousing technology and CATCH in the context of bioterrorism has been explored in proceedings of the IEEE International Conference on Intelligence and Security Informatics [6, 8]. In addition, this research effort received attention from various newspapers in Florida.

Contribution

The CATCH data warehouse research resulted in architecture and applications. This artifact was used effectively to collect data in a consistent and automated fashion from disparate local health-care organizations, which, among themselves, had no consistent IS or data collection infrastructure. The immediate contribution of this research to public health policy was the ability to collect data that could be effectively used to formulate such policy within Florida, where it was implemented. In a broader context, the artifact could serve as a template for the implementation of such systems elsewhere. Furthermore, the architecture and applications could serve as a model for the development of similar systems, such as one that was developed for bioterrorism surveillance, to serve other public or business needs.

Case 2: A Software Reuse Measure Developed at MBA Technologies

MBA Technologies was a medium-sized Phoenix-based software developer that specialized in the development of business process and accounting systems; the company obtained high reuse in its software development by leveraging of existing components that were mapped to an enterprise-level model. The model and its components represented generic business solutions that could be customized to a specific set of requirements. The objective of this work by Rothenberger and Hershauer [42] was to develop a generic reuse metric for such an enterprise-level model-based software development environment and to apply the generic measure to the specifics of the organization. Figure 3 provides a summary of the research steps discussed below.

Objective-Centered Solution

In spring 1997, Rothenberger and Hershauer [42] wanted to conduct an in-depth case study on the reuse efforts at MBA Technologies that required the assessment of the

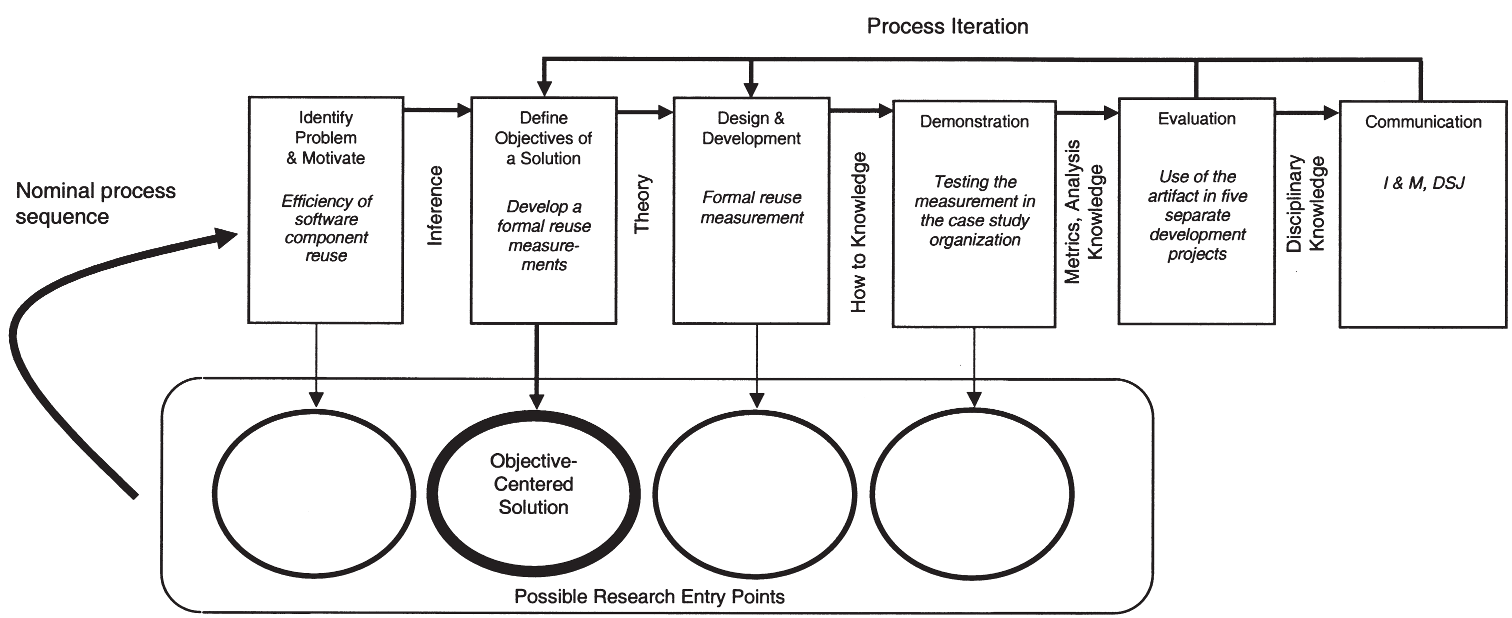

Figure 3. DSRM Process for the MBA Technologies Study

reuse rates the company obtained in its projects. Existing reuse measures available in the IS and computer science literatures were not suitable to assess the reuse rate in the enterprise-level model context; existing measures were only defined on a high level and did not define specifics required for an application to actual projects. To use the underlying principles of a high-level measure in the field setting, decisions had to be made about how to assess and count modified component reuse, partial component reuse, generated code, and multiple layers of abstraction.

Problem Identification and Motivation

Most software development companies do not assess their success at reuse, even if they are actively pursuing an increase in the reuse of software artifacts through a formal reuse program. Thus, many software developers invest in corporate reuse programs without being able to evaluate whether their programs lead to an increase of reuse. Also, without a formal reuse measure, they are not able to identify differences in reuse success among projects. The development and subsequent dissemination of a reuse measure that can be applied to enterprise-level model-based reuse efforts would enable the researchers to conduct an in-depth analysis of MBA Technologies' reuse success across multiple completed projects. Further, a measure would provide the means for continued monitoring of reuse success in software projects.

Objective of the Solution

The objective was to develop a reuse rate measure that allowed the researchers to assess the reuse rate, or reuse percentage, of the participating organization for subsequent case study research. Such a measure would represent the development effort that was reused from existing code as a percentage of the total project development effort. The measure was to be developed in a generic fashion that would ensure its applicability to settings other than the participating organization as well.

Design and Development

The software measurement literature was used to evaluate the suitability of potential size or complexity measures. The concept of the reuse rate was obtained from software reuse literature, which served as the theoretical foundation for the development of the reuse metric. The result of the design effort was a generic reuse measure that could be applied to any enterprise-level model-based reuse setting and that was customized to the specific organizational setting at MBA Technologies. The reuse rate was defined as the reused development effort divided by the total development effort of the project. The metric artifact operationalized this high-level definition by formalizing how to count reused development effort and total development effort in the context of an enterprise-level model-based reuse setting. This operationalization required making decisions on how to count duplicate use of code stubs, modified reused components, and other special cases. These decisions and assessments were made based on prior findings in the software reuse literature.

Demonstration

Assessing and reporting the reuse rate for a project in the participating organization demonstrated the measure's feasibility and efficacy. Details about the company's development environment, including a classification of code into three levels of abstraction, the use of generated code, specifics about the component design, and the classification of certain code stubs, were obtained through structured interviews. Size measures in thousands of lines of code (KLOC) and the classification of code stubs at the lowest level of abstraction were obtained directly from source code. The measure yielded separate reuse percentages for code on three layers of abstraction, according to the organization's classification, as well as a weighted total reuse percentage. Further, reused generated code was reported separately.

Evaluation

In the subsequent case study, the measure was used to assess the reuse rates of five projects at MBA Technologies, with sizes varying from 57 KLOC to 143 KLOC. The assessed total project reuse rate for nongenerated code ranged from 50.5 percent to 76.0 percent. In structured interviews, developers were asked to assess the projects' reuse rates without prior knowledge of the measured results. The relative assessments were consistent with the actual measurements.

Communication

The contributions of this effort were disseminated in peer-reviewed scholarly publications. The development of the reuse rate measure was published in *Information & Management* [42]. Further, the measure was used to assess the projects of the software development organizations in a subsequent case study that appeared in *Decision Sciences* [41].

Contribution

The research artifacts resulting from this study included a designed and evaluated formal measure and metric for software reuse rates. These artifacts provide a valid and effective measure for use in development practice at the organizational and project level for evaluation and assessment of the effectiveness and performance of software reuse efforts. They could be valuable measures for use in research where measures of software reuse are required.

Case 3: SIP-Based Voice- and Video-Over IP Software

The session initiation protocol (SIP) is an Internet Engineering Task Force (IETF) standard for Internet protocol (IP) telephony that was developed for voice-over Internet communication. Researchers at the Network Convergence Lab (NCL) at Claremont Graduate University were involved since early 2000 in the standardization process for SIP-based voice communication. In early 2001, the Internet2 Consortium wanted to explore video-over IP applications as an emerging architecture for IP networks in cooperation with the NCL. This led to a research effort by Chatterjee [9], Gemmill et al. [17], and Tulu et al. [47, 48] focusing on the extension of the SIP standard [9, 17, 47, 48], which is summarized in Figure 4.

A Design- and Development-Centered Approach

Building on the SIP-based voice communication standard, researchers at NCL aimed to design and deploy a voice- and videoconferencing-over IP application that enhances the SIP-based voice communication standard. This DS research artifact was to be deployed across 202 universities.

Problem Identification and Motivation

There were three particular technical problems that emerged in discussions within the IETF and the Internet2 consortium. First, while SIP standards were emerging, there were no actual SIP-based software artifacts that would provide telephony and video functionalities and features. Second, because universities and companies use a variety of vendor products, technologies, and standards, there was a need to develop middleware that provided a uniform way for storing and finding information related to video and voice users, as well as devices and technologies in enterprise directories. This problem was particularly relevant to Internet2 because universities were implementing diverse technology solutions. Third, there was a need to solve the security problem: some applications including SIP cannot traverse firewalls and fail to work when private IP addresses are used behind network address translators (NATs).

Objectives of the Solution

Several requirements drove the research effort. First, researchers needed to follow SIP technical standards closely. Second, the performance of the artifact could not be allowed to overwhelm the capabilities of a typical desktop computer of the time. Also, there were functions and features necessary to meet the requirements of the end users, including point-to-point calls, instant messaging, and videoconferencing. Furthermore, the middleware software for storing user and device information had to be compatible with existing directory services within participating campuses. Finally, a security solution was required that would be implemented within the application in such a way that no external measures were required within firewalls and routers.

Design and Development

The design and development process followed that of an IS development research project. It started with a requirements-gathering process, in which a diverse set of potential end users participated, resulting in requirements documentation, which was later used for designing a detailed technical architecture through Internet2 member meetings and

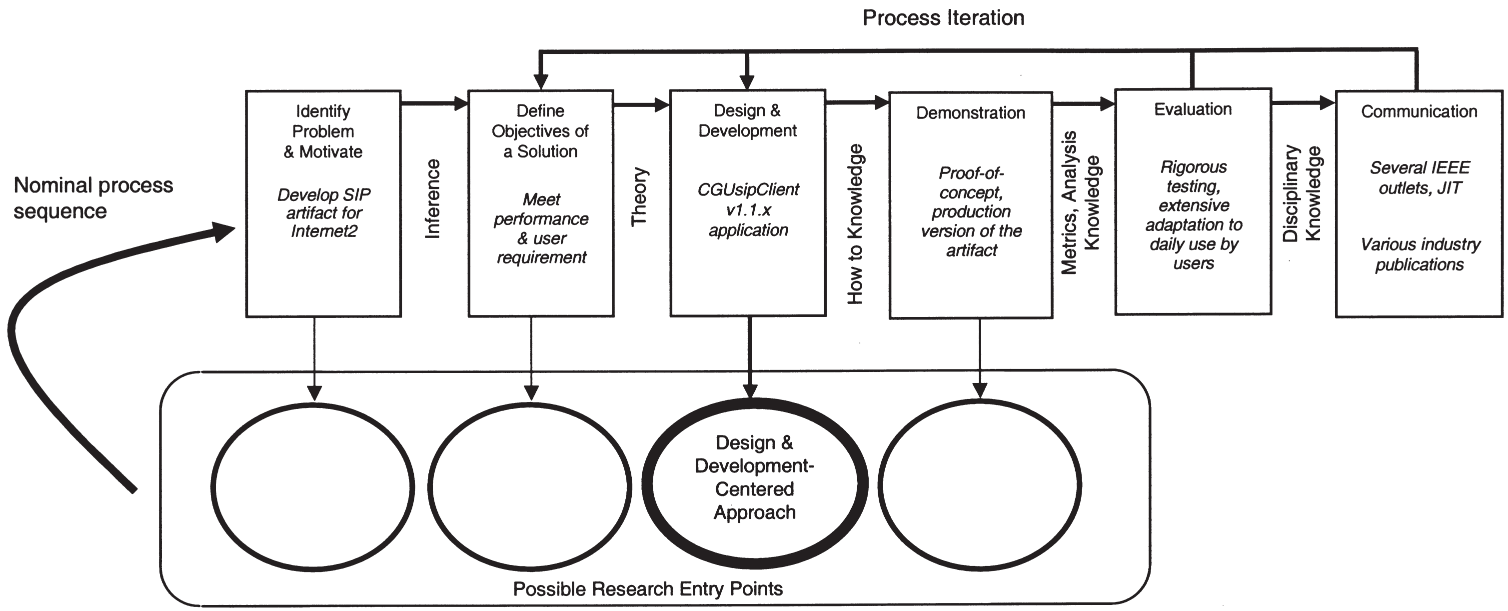

Figure 4. DSRM Process for the CGUsipClient v1.1.x Project

mailing list discussions. In particular, the middleware that was developed was standardized through the International Telecommunications Union's standardization section (ITU-T), which required participation from several European and other international participants. The software was developed, based on computer science and networking literature, to provide a proof-of-concept and a fully working client application.

Demonstration

The implemented artifact includes the SIP application and its directory middleware. Implementation details serve as a demonstration of the approach. CGUsipClient v1.1.x is a Java-based application implemented on a commercial SIP stack. It uses Java Media Framework (JMF) application programming interfaces (API) for voice and video operations. It provides point-to-point telephony, video calls, directory service lookup, click-to-call, and secured authentication. It uses technologies to solve the security problems mentioned above and utilizes a lightweight directory access protocol (LDAP)–based solution for providing directory information. It uses an H.350 directory to offer "white page," "click-to-call," and "single sign-on" facilities. White page displays user information. Click-to-call enables a user to call another user by clicking on the other user's SIP uniform resource identifier (URI). Single sign-on provides an authenticating facility with an SIP-based proxy or a registrar based on the credentials fetched from the LDAP structure instead of explicitly providing the user name and the password for registration.

Evaluation

Once the software was developed, researchers started a thorough testing process. First, the artifact was extensively tested for debugging purposes within a closed group. Next, the application was shared with the entire Internet2 community via a Web portal, where users were able to download the software after providing information about themselves. The information provided was automatically linked to the middleware directory. More than 250 institutions downloaded the software artifact. The researchers found that 30 percent of those who downloaded the software used it for one or more hours daily. In addition, the CGUsipClient was tested for performance, usability, and usefulness. The researchers measured the call setup time, CPU usage, end-to-end delay; all results were satisfactory. The test of the H.350 middleware standard implementation showed that the directory service performed well and lookup time was satisfactory. Finally, the security mechanism that was developed to open and close pinholes in the firewall/NAT for active SIP sessions was successfully tested, indicating only minimal and acceptable delays. The Internet2 working group was pleased with the efforts, judging the design process successful.

Communication

Preliminary results of this project were reported in refereed conferences (e.g., [47, 48]) and detailed results appeared in *IEEE Journal on Selected Areas in Communications* [9] and *Journal of Internet Technology* [17]. In addition, the middleware work received recognition in Internet2 and National Science Foundation (NSF) press releases. Trade magazines, such as *Network World,* and corporations, such as Packetizer, maintain the H.350 middleware standard information and details.

Contribution

This research enhanced and further developed the existing SIP voice-over IP standard into an SIP-based video-over IP standard. The enhanced standard was successfully evaluated and made available to the Internet2 working group, which may result in the commercial use of this new standard. Further, the existence of a video-over IP standard may serve as a foundation for future research aimed at enhancements of this technology.

Case 4: Developing a Method at Digia to Generate Ideas for New Applications That Customers Value

Digia Ltd. was a Helsinki-based research and development firm specializing in innovative software applications for wireless communication that focused on the creation of personal communication technologies and applications for wireless information devices. This case reports the efforts of Peffers and colleagues [34, 36] at Digia, also illustrated in Figure 5, to develop a better IS planning method.

A Client-Initiated Project

In fall 2000, Digia Chairman Pekka Sivonen approached one of the authors with a request to help define a portfolio of potential applications for Digia to develop to meet the need for financial services delivered by the next-generation wireless devices [36]. The researcher accepted this invitation because it fit with his current research objective and that of colleagues: to develop a method to support the generation of ideas for IS projects that would provide the greatest impact on achieving a firm's strategic goals. Because few applications for providing financial services using mobile devices were in operation at the time, this looked like a good opportunity to use a new conceptual method for IS planning that the authors earlier trialed in a business case environment. Because the client's objective was a portfolio of applications and the research objective was the development of a requirements engineering methodology for determining this portfolio, this meant that the initiative for the project came from a proposed demonstration of the new methodology.

Problem Identification and Motivation

Literature had shown that in most firms, there was no shortage of ideas for new IS projects, but most tended to be suboptimal [36]. The problem was to design a method for managers to make use of the ideas of many people within and around the organization, while keeping the focus on what is important and valuable for the firm. Bottom-up planning generates so many ideas that it may be impossible to sort out the few that

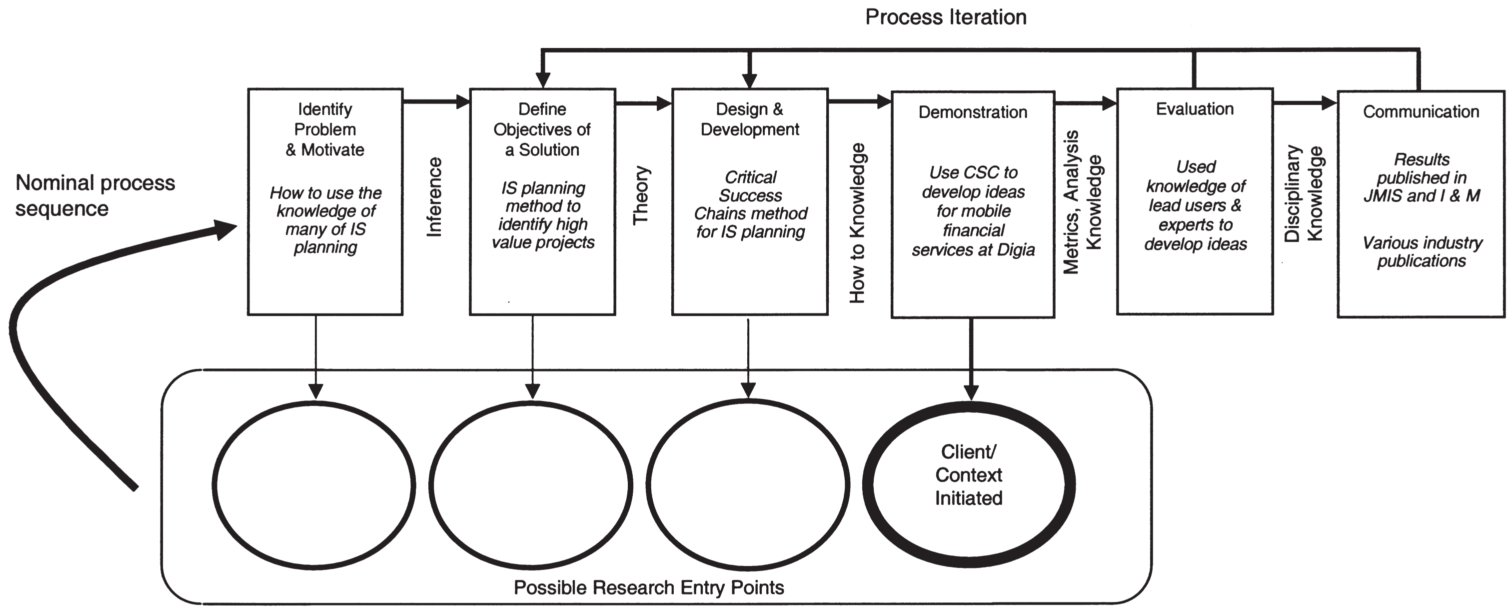

Figure 5. DSRM Process for the Digia Study

have the potential to have a high impact on the firm, because most are self-serving, narrowly focused, and of little potential impact. Top-down planning has the benefit of strategic perspective and better alignment with the interests of owners, but its weakness is an inability to take advantage of knowledge from around the organization and beyond the organization about ideas that may be important to the firm. It generally ignores all ideas except those that originate in the executive suites.

Objectives of the Solution

The researchers' objective was to demonstrate a new IS planning method in an industry setting. This allowed the researchers to study how well, in a noncontrolled test environment, the method would meet the proposed objectives: (1) allow them to make use of the ideas of many from in and around the organization; (2) include experts outside the firm and potential users, but keep the focus on ideas with high strategic value to the firm; and (3) transform the resulting data into forms that can be used for IS planning and development.

Design and Development

The Digia project researchers made use of a pilot study, conducted at Rutgers University [36], as the basic template for the new method. They used personal construct theory (PCT) [26] and critical success factors [39] as theoretical bases for the method development. For the data collection, they borrowed "laddering," a PCT-based technique developed for use in marketing research for structured interviewing, to collect rich data on subject reasoning and preferences. For the analysis, they adapted hierarchical value maps, which had been used in marketing to display aggregated laddering data graphically. They incorporated an ideation workshop, where business and technical expertise was brought to bear on the task of developing feasible ideas for new business applications from the graphical presented preferences and reasoning of the subjects. The result of the design effort was the CSC method for using the ideas of many people in and around the organization to develop portfolios of feasible application ideas that are highly valuable to the organization. In the Digia case, these concepts were applied to a real industry setting, which, in turn, allowed the researchers to extend the CSC method with concepts relevant to the case organization.

Demonstration

The researchers used the opportunity at Digia to demonstrate CSC's feasibility and efficacy [34, 36]. They started by recruiting and interviewing 32 participants, approximately evenly divided between experts and potential end users. They conducted individual structured interviews, using stimuli collected from the subjects ahead of time. The interview method was intended to encourage participants to focus on the value of ideas.

The laddering interviews provided rich data about applications the participants wanted and why. Using qualitative clustering, data were used to create five graphical maps, containing 114 preference and reasoning constructs. The next step was to conduct an ideation workshop with six business and engineering experts and managers from the firm to convert the participant preferences to feasible business project ideas at a "back-of-the-envelope" level. In the workshop, conducted in isolation in a single five-hour stretch, the participants developed three business ideas, with application descriptions, business models, and interaction tables. These were further developed by analysts in postworkshop work to be integrated into the firm's strategic planning effort.

Evaluation

The CSC method met the project's objectives. It enabled the researchers to use rich data collected by a widely representative sample of experts and potential lead users from outside the firm, to keep the focus on ideas of potential strategic importance to the firm, and to analyze the data in such a way so as to make it useful for IS planning in the firm. Digia representatives were enthusiastic about the results of the workshop [36]. This feedback and the successful implementation of the method in practice enabled the project researchers to present initial "proof-of-concept"-level validation of the new method [34, 36]. The firm intended to use the resulting applications to plan its continued product development efforts.

Communication

The case study was reported in the *Journal of Management Information Systems* [36] and *Information & Management* [34]. The structure of these papers closely follows the nominal sequence of activities presented in the DSRM. In addition, the findings were presented in several practitioner-oriented outlets, including a book chapter [35], technical reports (e.g., [49]), and trade magazine articles (e.g., [50]).

Contribution

The artifact developed as a result of this research is a method for IS planning that can be used to make use of the knowledge of many people from in and around the organization, maintain focus on potential systems and applications of strategic value to the firm, and produce outputs of use to designers and managers in the IS development process. Consequently, it may be valuable for use in IS planning practice. In research, the method may be extended to enable planning efforts that focus on the development of requirements at the feature level, particularly to develop requirements engineering methods for such contexts as the development of cross-cultural feature sets and for special populations, such as for disabled persons.

Evaluation of the DSRM Process

We evaluate the DSRM process in terms of the three objectives for the DSRM described above. First, it should be consistent with prior DS research theory and practice, as it has been represented in the IS literature, and with design and DS research, as it has been conveyed in representative literature in reference disciplines. Second, it should provide a nominal process for conducting DS research in IS. Third, it should provide a mental model for the characteristics of research outputs. We will address each objective below.

First, the DSRM process is consistent with concepts in prior literature about DS in IS. Because we used a consensus method to design the DSRM, this consistency is an inherent outcome of the process. For example, Nunamaker et al.'s [33] five-step methodology can be mapped roughly to the DSRM process. Likewise, Walls et al.'s [55, 56] "components of an information system design theory," Takeda et al.'s [46] "design cycle" solution for intelligent computer-aided design systems, Rossi and Sein's [40] steps, Archer's [2] process for industrial design, Eekels and Roozenburg's [14] process for engineering design, and Hevner et al.'s [20] guidelines for the required elements of design research are all consistent with the DSRM.

Second, the DSRM provides a nominal process for conducting DS research. In addition, in the demonstration of four cases, we showed how each of the four DS research projects described in the cases followed a process consistent with the DSRM. In addition, the cases demonstrate each of the four research entry points described in the DSRM, including a problem-centered initiation, an objective-centered initiation, a design- and development-centered initiation, and a client-/context-centered initiation. In each case, the process worked well, and it was effective for its intended purposes.

Third, the DSRM provides a mental model for the presentation of research outcomes from DS research. The explication of the CATCH data warehouse, reported in Berndt et al. [5], incorporated all of the elements of the DSRM process, although it did not use the DSRM terminology. Rothenberger and Hershauer [42] followed the general outlines of the process in the structure of the paper, including a statement of the problem in the introduction, an explicit "purpose" section to outline the objectives of a solution, a design section called "creating the measure," demonstration sections called "application of the measure to a specific problem," and "example case data." Peffers et al. used a structure based on Hevner et al. [20] and consistent with the DSRM to report the Digia case [36]. Chatterjee et al. [9] incorporated elements of the DSRM in presenting the research. The paper identified the problem and defined the potential objectives or "benefits" of a solution in the introduction; it also incorporated the other elements of the DSRM in "Design, Implementation, and Performance of CGUsipClient" [9, p. 1924].

Discussion

We defined design science earlier in this paper. Recently, however, researchers [9] have raised questions about similarities between DS and action research. Both Cole et al. [10] and Järvinen [24] concluded that the similarities between these research approaches are substantial. Cole et al. [10] argued that the approaches share important assumptions regarding ontology, epistemology, and axiology. Järvinen [24] pointed to many similarities, although they employ different terminology, and went so far as to suggest that we cannot clearly differentiate between them. Perhaps the clearest distinction between them is found in their conceptual origins. DS research comes from a history of design as a component of engineering and computer science research, while action research originates from the concept of the researcher as an "active participant" in solving practical problems in the course of studying them in organizational contexts. In DS research, design and the proof of its usefulness is the central component, whereas in action research, the focus of interest is the organizational context and the active search for problem solutions therein. Resolution of this point will have to remain outside the scope of this paper, but it presents an interesting and perhaps fruitful area for further thought. It would appear that the DSRM could be used as a structure to present action research. Likewise, the search for a designed artifact could be presented as action research. Clearly the side-by-side existence of the two methodologies presents the researcher with choices for the structure of the research process and the presentation of the resulting solution. This discussion also raises an interesting question about whether the DSRM could be used in an action research study, whether researchers could use it to design new innovations based on technical, social, or informational resources or their combinations [24], and whether action research and DS research could be conceptually and methodologically integrated.

The DSRM is intended as a methodology for research; however, one might wonder whether it might also be used as a methodology for design in practice. There would appear to be no reason it could not be so used; however, there are elements of the DSRM that are intended to support essential DS research characteristics that might not always apply well to design in practice.

A design artifact, such as a curved wooden staircase, a kitchen appliance, or a surgical knife, is not necessarily required to embody new knowledge that would be conveyed to an audience through a scientific publication outlet. Consequently, there is no inherent requirement that a designer employ any rigorous process to create it. There may, on the other hand, be organizational, evidentiary, regulatory, or other reasons why some level of process rigor may be required. The designer of the curved staircase might be free to work from a simple sketch with a few measurements, while the designer of the surgical knife might be required to proceed through a careful process of data collection, consultation, documentation, and testing. Thus, for design in practice, the DSRM may contain unnecessary elements for some contexts, while being much too general to support design in others.

An important step in the evaluation of the DSRM was its application to four cases of previously published DS research. The four studies were chosen in part because they represent examples of DS research with four different entry points as specified in the DSRM. None of the articles in which these studies were reported used the language of the DSRM to explain its research approach. Instead, they each used ad hoc arguments to support the validity of the research. We found this to be common in prior DS research in the IS field, because hitherto no generally accepted framework for conducting and presenting DS research existed, at least not until Hevner et al.'s [20] guidelines provided characteristics of good DS research outcomes. Like Hevner et al. [20], we have used secondary data, in the form of four cases, to demonstrate the application of the DSRM. Results of the analysis of the four cases show that they are all instances of DS research that can be well framed in terms of the DSRM. Thus, we used the case discussions as a vehicle not only to evaluate the DSRM but also to transfer established DS research into a formal research framework and to illustrate its applicability. We expect that the case studies will provide useful templates for researchers who want to apply DSRM to their efforts. The development and evaluation of the DSRM was heavily influenced by design research, thus DSRM concepts have guided us in the conduct and presentation of this work and this is reflected in the structure of this paper. Clearly, the next step would be to directly adopt the proposed methodology in new DS research. This is something that we are currently working on in our ongoing research.

The ad hoc justification of prior DS research suggests the difficulty that authors faced, for lack of reference to a commonly accepted DS methodology, in supporting the validity of DS research in IS. Without a framework that is shared by authors, reviewers, and editors, DS research runs the danger of being mistaken for poor-quality empirical research or for practice case study. The DSRM completes a DS research paradigm with a methodology that is consistent with the DS research processes employed in the IS discipline, in this way establishing a common framework for future researchers to validate DS research, without making ad hoc arguments for its validity.

Conclusion

In this paper, we sought to develop a methodology for DS research in IS. We wanted this methodology to be well grounded in existing literature about DS in IS and related disciplines. In addition, we wanted a methodology that would provide guidance for researchers who work on DS research and provide a mental model for the presentation of its outcomes.

Interestingly, other research paradigms have been adapted for use in our discipline without such a formal definition. This is hardly surprising because IS research, if one counts from the tenure of our senior journals, is only about one-third of a century old. As a result, it was handed behavioral and natural science traditions from much older research disciplines in the business academe and adopted them without much adaptation. Consequently, this paper represents a unique effort to formally define a research methodology for use in IS.

We should emphasize that this paper represents one general methodological guideline for effective DS research. Researchers should by no means draw any inference that the DSRM is *the* only appropriate methodology with which to conduct such research. We can imagine that the efforts of others could result in at least five other types of DSRM:

1. A methodology to support curiously motivated DS research, although such research is not common in business disciplines, might look quite different than the DSRM. Some research in the social and natural sciences is driven primarily by curiosity and may therefore lack explicit outcome objectives.

2. A methodology to support research within a specific stream in IS might incorporate elements specific to the context of that research. For example, a methodology to support the design of methods for requirements analysis might provide guidelines for specific expected elements of requirements analysis, including organizational context, data gathering, modeling, and the form of the requirements specification. We observed a number of context-specific design research methodologies in engineering.
3. Whereas the motivation for the research is to solve problems in a specific organizational context, action research, as suggested in preceding paragraphs, may be an alternative or complementary paradigm through which to design IS research artifacts.
4. With respect to specific activities in the research process, future researchers may enhance the DSRM, for example, by developing subsidiary processes.
5. Finally, circumstances, such as context-specific constraints, may motivate researchers to develop and implement ad hoc processes that, while inconsistent with this DSRM, may, nonetheless, be well justified and produce valid results.

While these five examples come readily to mind, it seems likely that there are other ways that DS research could be well done. We present these alternatives here, without recommendation and without knowledge of their prior use, in speculation about what valid alternatives to the DSRM might be subsequently developed and used. In doing so, we are suggesting that the DSRM should not be used as a rigid orthodoxy to criticize work that does not follow it explicitly.

The case studies we provided with this paper demonstrate its use within the scope of four research problems. Further use will tell us whether there are problem domains where it requires extension or where it does not work well. Another interesting problem is that of the research entry point. We demonstrated that there are multiple possible entry points for DS research. Of course, this issue is not unique to DS research. We do not recall reading a theory-testing paper where the authors say that they decided on the research questions after they collected the data or even after they did the analysis, but we have all observed that this happens with no ill effects. The "scientific method" is an espoused theory, approximated but not always matched by theory in use. We think that a research methodology should account, as far as it is practical, for the research process in use.

Acknowledgments: The first two authors made substantially similar contributions to this paper. First authorship was determined by rotation among papers.

References

1. Adams, L., and Courtney, J. Achieving relevance in IS research via the DAGS framework. In R.H. Sprague Jr. (ed.), *Proceedings of the Thirty-Seventh Annual Hawaii International Conference on System Sciences.* Los Alamitos, CA: IEEE Computer Society Press, 2004 (available at http://csdl2.computer.org/persagen/DLAbsToc.jsp?resourcePath=/dl/proceedings/&toc=comp/proceedings/hicss/2004/2056/08/2056toc.xml&DOI=10.1109/HICSS.2004.1265615).

2. Archer, L.B. Systematic method for designers. In N. Cross (ed.), *Developments in Design Methodology.* London: John Wiley, 1984, pp. 57–82.

3. Aumann, H.H.; Chahine, M.T.; Gautier, C.; Goldberg, M.D.; Kalnay, E.; McMillin, L.M.; Revercomb, H.; Rosenkranz, P.W.; Smith, W.L.; Staelin, D.H.; Strow, L.L.; and Susskind, J. AIRS/AMSU/HSB on the Aqua mission: Design, science objectives, data products, and processing systems. *IEEE Transactions on Geoscience and Remote Sensing, 41,* 2 (2003), 253–264.

4. Berndt, D.J.; Hevner, A.R.; and Studnicki, J. Data warehouse dissemination strategies for community health assessments. *Upgrade, 2,* 1 (2001), 48–54.

5. Berndt, D.J.; Hevner, A.R.; and Studnicki, J. The CATCH data warehouse: Support for community health care decision-making. *Decision Support Systems, 35,* 3 (2002), 367–384.

6. Berndt, D.J.; Hevner, A.R.; and Studnicki, J. Bioterrorism surveillance with real-time data warehousing. In H.E.A. Chen (ed.), *IEEE International Conference on Intelligence and Security Informatics.* Los Alamitos, CA: IEEE Computer Society Press, 2003, pp. 322–335.

7. Berndt, D.J.; Fisher, J.W.; Hevner, A.R.; and Studnicki, J. Healthcare data warehousing and quality assurance. *IEEE Computer, 34,* 12 (December 2001), 56–63.

8. Berndt, D.J.; Bhat, S.; Fisher, J.W.; Hevner, A.R.; and Studnicki, J. Data analytics for bioterrorism surveillance. In H.E.A. Chen (ed.), *IEEE International Conference on Intelligence and Security Informatics.* Los Alamitos, CA: IEEE Computer Society Press, 2004, pp. 17–27.

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

10. Cole, R.; Purao, S.; Rossi, M.; and Sein, M.K. Being proactive: Where action research meets design research. In D. Avison, D. Galletta, and J.I. DeGross (eds.), *Twenty-Sixth International Conference on Information Systems.* Atlanta: Association for Information Systems, 2005, pp. 325–336.

11. Cooper, R.G. Stage-Gate Systems—A new tool for managing new products. *Business Horizons, 33,* 3 (1990), 44–54.

12. Cooper, R.G. Winning with new products: Doing it right. *IVEY Business Journal, 64,* 6 (July–August 2000), 54–60.

13. DMReview. Glossary. SourceMedia, Brookfield, WI, 2007 (available at www.dmreview.com/glossary/a.html).

14. Eekels, J., and Roozenburg, N.F.M. A methodological comparison of the structures of scientific research and engineering design: Their similarities and differences. *Design Studies, 12,* 4 (1991), 197–203.

15. Evbuonwan, N.F.O.; Sivaloganathan, S.; and Jebb, A. A survey of design philosophies, models, methods and systems. *Proceedings Institute of Mechanical Engineers,* 210 (1996), 301–320.

16. Fulcher, A.J., and Hills, P. Towards a strategic framework for design research. *Journal of Engineering Design, 7,* 1 (1996), 183–193.

17. Gemmill, J.; Srinivasan, A.; Lynn, J.; Chatterjee, S.; Tulu, B.; and Abhichandani, T. Middleware for scalable real-time multimedia communications cyberinfrastructure. *Journal of Internet Technology, 5,* 4 (2004), 405–420.

18. Gentner, D., and Stevens, A.S. *Mental Models.* Hillsdale, NJ: Lawrence Erlbaum, 1983.

19. Haran, M.; Karr, A.; Last, M.; Orso, A.; Porter, A.; Sanil, A.; and Fouche, S. Techniques for classifying executions of deployed software to support software engineering tasks. *IEEE Transactions on Software Engineering, 33,* 5 (2007), 287–304.

20. Hevner, A.R.; March, S.T.; and Park, J. Design research in information systems research. *MIS Quarterly, 28,* 1 (2004), 75–105.

21. Hickey, A.M., and Davis, A.M. A unified model of requirements elicitation. *Journal of Management Information Systems, 20,* 4 (Spring 2004), 65–84.

22. Hoffman, R.R.; Roesler, A.; and Moon, B.M. What is design in the context of human-centered computing? *IEEE Intelligent Systems, 19,* 4 (2004), 89–95.

23. Iivari, J.; Hirschheim, R.; and Klein, H.K. A paradigmatic analysis contrasting information systems development approaches and methodologies. *Information Systems Research, 9,* 2 (1998), 164–193.

24. Järvinen, P. Action research is similar to design science. *Quality & Quantity, 41,* 1 (2007), 37–54.

25. Johnson-Laird, P., and Byrne, R. A gentle introduction. Mental Models Website, School of Psychology, Trinity College, Dublin, 2000 (available at www.tcd.ie/Psychology/Ruth_Byrne/mental_models/).

26. Kelly, G.A. *The Psychology of Personal Constructs.* New York: W.W. Norton, 1955.

27. Klassi, J. Environmental enhancement of the oceans by increased solar radiation from space. *Oceans, 17* (November 1985), 1290–1295.

28. Krishnamurthy, D.; Rolia, J.A.; and Majumdar, S. A synthetic workload generation technique for stress testing session-based systems. *IEEE Transactions on Software Engineering, 32,* 11 (2006), 868–882.

29. Lisetti, C., and LeRouge, C. Affective computing in tele-home health. In R.H. Sprague Jr. (ed.), *Proceedings of the Thirty-Seventh Annual Hawaii International Conference on System Sciences.* Los Alamitos, CA: IEEE Computer Society Press, 2004 (available at http://csdl2.computer.org/persagen/DLAbsToc.jsp?resourcePath=/dl/proceedings/&toc=comp/proceedings/hicss/2004/2056/06/2056toc.xml&DOI=10.1109/HICSS.2004.1265373).

30. Maguire, M. Methods to support human-centered design. *International Journal of Human–Computer Studies, 55,* 4 (2001), 587–634.

31. March, S., and Smith, G. Design and natural science research on information technology. *Decision Support Systems, 15,* 4 (1995), 251–266.

32. McPhee, K. Design theory and software design. Technical Report, Department of Computing Science, University of Alberta, 1996.

33. Nunamaker, J.F.; Chen, M.; and Purdin, T.D.M. Systems development in information systems research. *Journal of Management Information Systems, 7,* 3 (Winter 1990–91), 89–106.

34. Peffers, K., and Tuunanen, T. Planning for IS applications: A practical, information theoretical method and case study in mobile financial services. *Information & Management, 42,* 3 (2005), 483–501.

35. Peffers, K., and Tuunanen, T. The process of developing new services. In T. Saarinen, M. Tinnilä, and A. Tseng (eds.), *Managing Business in a Multi-Channel World: Success Factors for E-Business.* Hershey, PA: Idea Group, 2005, pp. 281–294.

36. Peffers, K.; Gengler, C.; and Tuunanen, T. Extending critical success factors methodology to facilitate broadly participative information systems planning. *Journal of Management Information Systems, 20,* 1 (Summer 2003), 51–85.

37. Preston, M., and Mehandjiev, N. A framework for classifying intelligent design theories. In N. Mehandjiev and P. Brereton (eds.), *Proceedings of the 2004 ACM Workshop on Interdisciplinary Software Engineering Research.* New York: ACM Press, 2004, pp. 49–54.

38. Reich, Y. The study of design methodology. *Journal of Mechanical Design, 117,* 2 (1994), 211–214.

39. Rockart, J.F. Chief executives define their own data needs. *Harvard Business Review, 57,* 2 (1979), 81–93.

40. Rossi, M., and Sein, M.K. Design research workshop: A proactive research approach. Paper presented at the Twenty-Sixth Information Systems Research Seminar in Scandinavia, Information Systems Research in Scandinavia Association, Haikko, Finland, August 9–12, 2003.

41. Rothenberger, M.A. Project-level reuse factors: Drivers for variation within software development environments. *Decision Sciences, 34,* 1 (2003), 83–106.

42. Rothenberger, M.A., and Hershauer, J.C. A software reuse measure: Monitoring an enterprise-level model driven development process. *Information & Management, 35,* 5 (1999), 283–293.

43. Simon, H. *The Sciences of the Artificial.* Cambridge, MA: MIT Press, 1969.

44. Studnicki, J.; Steverson, B.; Myers, B.; Hevner, A.; and Berndt, D. Comprehensive assessment for tracking community health (CATCH). *Best Practices and Benchmarking in Healthcare, 2,* 5 (1997), 196–207.

45. Swaab, R.I.; Postmes, T.; Neijens, P.; Kiers, M.H.; and Dumay, A.C.M. Multi-party negotiation support: Visualization and the development of shared mental models. *Journal of Management Information Systems, 19,* 1 (Summer 2002), 129–150.

46. Takeda, H.; Veerkamp, P.; Tomiyama, T.; and Yoshikawam, H. Modeling design processes. *AI Magazine, 11,* 4 (Winter 1990), 37–48.

47. Tulu, B.; Chatterjee, S.; and Laxminarayan, S. A taxonomy of telemedicine efforts with respect to applications, infrastructure, delivery tools, type of setting and purpose. In R.H. Sprague Jr. (ed.), *Proceedings of the Thirty-Eighth Annual Hawaii International Conference on System Sciences.* Los Alamitos, CA: IEEE Computer Society Press, 2005 (available at http://csdl2.computer.org/comp/proceedings/hicss/2005/2268/06/22680147b.pdf).

48. Tulu, B.; Abhichandani, T.; Chatterjee, S.; and Li, H. Design and development of an SIP-based video conferencing application. In M.M. Freire, P. Lorenz, and M.M.-O. Lee (eds.), *High Speed Networks and Multimedia Communications.* Heidelberg/Berlin: Springer, 2003, pp. 503–512.

49. Tuunanen, T. Critical success chains method. Technical Report LTT-Tutkimus Oy, Elektronisen Kaupan Instituutti, Helsinki, 2001.

50. Tuunanen, T. Mitä käyttäjät todella haluavat [What do the users really want?]. *Tietoviikko* (Helsinki), 2002, 1.

51. Vaishnavi, V., and Kuechler, B. Design research in information systems. Association for Information Systems, 2005 (available at www.isworld.org/Researchdesign/drisISworld.htm).

52. van Aken, J.E. Management research based on the paradigm of the design sciences: The quest for field-tested and grounded technological rules. *Journal of Management Studies, 41,* 2 (2004), 219–246.

53. van Aken, J.E. Valid knowledge for the professional design of large and complex design processes. *Design Studies, 26,* 4 (2005), 379–404.

54. Vandenbosch, B., and Higgins, C.A. Executive support systems and learning: A model and empirical test. *Journal of Management Information Systems, 12,* 2 (Fall 1995), 99–130.

55. Walls, J.; Widmeyer, G.; and El Sawy, O. Building an information system design theory for vigilant EIS. *Information Systems Research, 3,* 1 (1992), 36–59.

56. Walls, J.; Widmeyer, G.; and El Sawy, O. Assessing information system design theory in perspective: How useful was our 1992 initial rendition? *Journal of Information Technology Theory & Application, 6,* 2 (2004), 43–58.