Hemant Jain
Atish P. Sinha
Padmal Vitharana (Eds.)

LNCS 6629

Service-Oriented Perspectives in Design Science Research

6th International Conference, DESRIST 2011
Milwaukee, WI, USA, May 2011
Proceedings

Springer

Lecture Notes in Computer Science 6629

Hemant Jain Atish P. Sinha
Padmal Vitharana (Eds.)

Service-Oriented Perspectives in Design Science Research

6th International Conference, DESRIST 2011
Milwaukee, WI, USA, May 5-6, 2011
Proceedings

Springer

Volume Editors

Hemant Jain
University of Wisconsin-Milwaukee, Sheldon B. Lubar School of Business
3202 N. Maryland Avenue, Milwaukee, WI 53211, USA
E-mail: jain@uwm.edu

Atish P. Sinha
University of Wisconsin-Milwaukee, Sheldon B. Lubar School of Business
3202 N. Maryland Avenue, Milwaukee, WI 53211, USA
E-mail: sinha@uwm.edu

Padmal Vitharana
Syracuse University, Whitman School of Management
721 University Avenue, Syracuse, NY 13244-2450, USA
E-mail: padmal@syr.edu

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-20632-0 e-ISBN 978-3-642-20633-7
DOI 10.1007/978-3-642-20633-7
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011925683

CR Subject Classification (1998): H.4, I.2, C.2, H.3, D.2, H.5

LNCS Sublibrary: SL 3 – Information Systems and Application, incl. Internet/Web and HCI

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

There has been a surge of interest in design science research in information systems in the last few years. The goal of the design science research paradigm is to extend the boundaries of human and organizational capabilities by designing new and innovative constructs, models, methods, processes, and systems. Scholars from different backgrounds—such as information systems, computer science, software engineering, and medical informatics—are actively engaged in generating novel solutions to interesting design problems in information systems.

With five successful conferences already held in Claremont, Pasadena, Atlanta, Philadelphia, and St. Gallen, the International Conference on Design Science Research in Information Systems and Technology (DESRIST) has become the premier venue for exchanging design science research ideas in the IS field. The sixth DESRIST conference in Milwaukee brought together researchers from all over the world, including Australia, Austria, Canada, Denmark, Finland, Germany, Liechtenstein, South Korea, Sweden, Switzerland, and the USA.

The topical theme of DESRIST 2011 was "Service-Oriented Perspectives in Design Science Research." Several papers in the conference proceedings conform to the theme, focusing on topics such as service design, service-oriented architecture design, design of financial services, etc. In addition to the papers addressing the conference theme, the proceedings include cutting-edge research in several other areas. Some papers address the need for theory development in design science, while others formulate design science research strategies and guidelines. There are several papers on design evaluation and design methods. Also, there are papers that present design research exemplars in domains such as mobile computing, telecommunications, social media, healthcare, and finance. Finally, a couple of papers address a topic that is at the frontiers of design research in IS: the use of neuroscience.

In all, 50 papers were submitted to the conference for review. Each paper was reviewed by at least two referees. The reviews were double blind, meaning that each of the two groups – authors and referees – remained anonymous to one another. Finally, 29 papers were selected to be published as full-length research papers, yielding an acceptance rate of 58%. Five other papers were selected to be published as short papers.

We thank all the authors who submitted papers to the DESRIST 2011 conference. We hope the readers will find the papers as interesting and informative as we did. We would like to thank all the members of the Program Committee, as well as the additional referees, who took the time to provide detailed and constructive reviews for the authors. We would also like to thank the other members of the Organizing Committee, as well as the volunteers, whose dedication and effort helped bring about a successful DESRIST 2011 conference. We believe that the papers in the DESRIST 2011 proceedings provide several interesting and valuable insights into the theory and practice of design science, as well as open up new and exciting possibilities for research in the discipline.

May 2011

Hemant Jain
Atish P. Sinha
Padmal Vitharana

Organization

DESRIST 2011 was organized by the Sheldon B. Lubar School of Business, University of Wisconsin-Milwaukee.

General Chairs

Robert Winter	University of St.Gallen, Switzerland
J. Leon Zhao	City University of Hong Kong, Hong Kong

Program Chairs

Hemant Jain	University of Wisconsin-Milwaukee, USA
Atish P. Sinha	University of Wisconsin-Milwaukee, USA

Proceedings Chair

Padmal Vitharana	Syracuse University, USA

Local Arrangements Chair

Mark Srite	University of Wisconsin-Milwaukee, USA

Publicity Chairs

Mark Toleman	University of Southern Queensland, Australia
Karthikeyan Umapathy	University of North Florida, USA

Program Committee

Ahmed Abbasi	University of Wisconsin-Milwaukee, USA
Stephan Aier	University of St.Gallen, Switzerland
Antonia Albani	Technical University of Delft, The Netherlands
Richard Baskerville	Georgia State University, USA
Udo Bub	Deutsche Telekom Laboratories, Germany
Tobias Bucher	Phoenix Group, Germany
Sven Carlsson	Lund University, Sweden
Samir Chatterjee	Claremont Graduate School, USA
Roger Chiang	University of Cincinnati, USA
Jae Choi	Middle East Technical University, Cyprus
Chen-Huei Chou	College of Charleston, USA

Additional Reviewers

Financial Support

Sheldon B. Lubar School of Business, University of Wisconsin-Milwaukee
J. Mack Robinson College of Business, Georgia State University

Table of Contents

Design Theory

Design Science Research Strategies

Design Methods and Techniques

Design Evaluation

Design Guidelines

Service-Oriented Perspectives in Design Science

Process Design

Neuroscience in Design Research

Designing for Social Media

Theorizing in Design Science Research

Jong Seok Lee[1], Jan Pries-Heje[2], and Richard Baskerville[1]

[1] Georgia State University, CIS Department, Atlanta, GA, 30303, USA
jlee@cis.gsu.edu, baskervile@acm.org
[2] Roskilde University, Department of Communication, Business & IT, Roskilde, Denmark
janph@ruc.dk

Abstract. Theory is a central element in research. Due to the importance of theory in research, considerable efforts have been made to better understand the process of theorizing, i.e., development of a theory. A review of the literature in this area suggests that two dominant theorizing approaches are anchored to deductive and inductive reasoning respectively. In contrast, an essential part of theorizing for design may involve abductive reasoning. The purpose of design theory is not to advance declarative logic regarding truth or falseness, but to guide learning and problem solving through the conceptualization of a design artifact. This paper critically examines the process of theorizing for design by developing an idealized design theorizing framework. The framework indicates that theorizing for design operates in two distinct domains: instance and abstract. Further, four key theorizing activities are identified in this framework: abstraction, solution search, de-abstraction, and registration. The framework provides grounds for building strong design theories in the design science paradigm by explicating the underlying theorizing process for design.

Keywords: Design Theory, Theorizing in Design Science Research.

1 Introduction

Design science research holds promise as a paradigm that can establish the relevance of academic information systems (IS) research for IS practice [1]. However, unless such research develops a solid contribution to theory, the paradigm loses its importance to academia [2]. While there is substantial work that describes design science theories [3-4], less is known about the process of creating theories in design science. If design theory is indeed a particular kind of theory, it follows that design theorizing may be a particular kind of theorizing. The purpose of this paper is to describe and illustrate an idealized process for theorizing in design science. Such theorizing processes are important in design science research if the paradigm is to maintain its contribution to the IS academic tradition while simultaneously making significant advances in IS practice.

The substantiation of a strong theoretical contribution is often regarded as *prima facie* evidence of high quality in scholarly work. However, definitions of "strong" theory, not to mention theory itself are so contentious among academics that it may be easier to exclude non-theory than it is to inclusively define theory [5]. Alternatively,

H. Jain, A.P. Sinha, and P. Vitharana (Eds.): DESRIST 2011, LNCS 6629, pp. 1–16, 2011.

a focus on the quality of the process of theorizing may be more meaningful than evaluating the quality of the theory under development [6]. Thus, the issues of the quality of the theory (as a product) are intertwined with the quality of the theorizing (as a process) because theorizing is critical in producing good theories; and necessarily to high quality research that contributes substantial theories. There are elaborations of inductive theorizing [7-10], deductive theorizing [11], and richer conceptions of theorizing as messy, human behavior [12]. Different research paradigms take on different theorizing approaches that lead to different types of theory, e.g., systematic, formal, or axiomatic [13].

Theory in design science research is deemed by many authorities to be so important that a distinct class of *design theory* is widely accepted [3-4]. These theories have specific components, for example, meta-requirements, a meta-design, a design method, testable product hypotheses, testable process hypotheses, etc. Design theories will typically encompass a design process for applying a meta-design for the purposes of instantiating a designed artifact. While there is an established body of work dedicated to explaining and understanding design theory and its components, there is a need for further examination of the process of theorizing for design. In general, the literature recognizes that theory and theorizing are intertwined, suggesting a need for more attention to design theorizing. Understanding the theorizing of design is important because it should help guide design science researchers to build stronger design theories.

Weick [12] recognizes that many discussions of design theorizing, like other theorizing processes, are rational idealizations of a disciplined form of imagination. The products of theorizing (theories) are social constructions that evolve from an ideation process of concurrent trials (conjectures) and errors (refutations). It is often a variant of other sense-making processes such as generalization, prediction, and problem solving. Unlike theory testing processes, theorizing is a search for plausibility rather than validity, and selecting one theory from among other imagined constructs may be because of its interest, believability, or beauty. Theorizing is rarely mechanistic, but is often a process characterized by an *"intuitive, blind, wasteful, serendipitous, creative quality"* (p. 519).

2 Deductive, Inductive, and Abductive Theorizing

Theorizing refers to the process of constructing a theory [12]. It is often described as interim struggles in which patterns that explain the relation of one property with another are searched and proposed, and the truth of such proposed patterns is examined through experience [6, 14]. Also, theorizing may be a form of *disciplined imagination* in which concurrent trial-and-error thinking is iterated through imaginary experiments [12]. Weick [12] suggests that theorizing largely consists of three components: problem statements, thought trials, and selection criteria. It is a process that involves concurrency and iteration in each of these components. Kaplan [15] made a distinction between knowledge growth by extension and knowledge growth by intension. Knowledge growth by extension concerns exploring new areas by applying the existing knowledge in one area to adjacent areas, whereas knowledge growth by intention concerns seeking more complete knowledge that operates within a single area. The assumption underlying these two theorizing strategies is the intellectual reasoning method; knowledge growth by extension corresponds to inductive reasoning in which new knowledge is explored, whereas knowledge growth by intention corresponds to deductive reasoning in which existing knowledge is refined and tested. Inductive and deductive reasoning have been two dominant theorizing approaches in many research disciplines.

The origin of deductive reasoning dates back to ancient philosophy; Plato denied the validity of inductive sense making from experience, and asserted that only logical deduction is a valid method for developing theory, i.e., the hypothetico-deductive method. Deductive theorizing involves deducing a conclusion from a general premise, i.e., a known theory, to a specific instance (i.e., an observation). For instance, (a) premise: failure to incorporate user requirements leads to low user satisfaction, (b) instance: a system has failed to incorporate user requirements, (c) conclusion: the users of this system have low satisfaction. At the heart of deductive reasoning is *falsification* which suggests that a theory can only be shown to be wrong, but never be proven to be right [11]. Theorists using a deductive approach deduce hypotheses from general knowledge and attempt to falsify them in a variety of settings; thus, a surviving theory is deemed to become more complete.

In contrast, Aristotle recognized inductive reasoning as a valid method for generating knowledge, proceeding from particulars to generals. Bacon later conceptualized inductive reasoning by arguing that a theory can be inductively developed through discovering essential nature of observations. Inductive theorizing involves drawing a conclusion from specific instances. For instance, (a) instance: every system that failed to incorporate user requirements has resulted in low user satisfaction, (b) conclusion: failure to incorporate user requirements leads to low user satisfaction. Inductive theorizing is recognized a valid theorizing method by modern researchers [7-9, 16].

While the literature on deductive and inductive reasoning crystallizes two contrasting ways in which researchers can approach theorizing, Weick [12] criticizes such methodical views on theorizing, claiming that the process of theorizing is depicted as *mechanistic* when in fact it is intuitive and creative thinking process. Weick further argues that theorizing should be seen as sense making in that it involves a searching process where explanatory relationships are sought in concepts observed in the real world [17-18]. Theorizing may go beyond just a mechanistic approach based on deductive or inductive reasoning when it indeed involves making sense out of a phenomenon in a complex and open system. Furthermore, the product that comes out of theorizing may not always be a singular truth, but rather a situated truth that explains the given phenomenon well enough per human's intuition and creativity.

Simon [19] associated design logic with imperative logic, contrasting this with the declarative logic that inhabits both inductive and deductive reasoning. Recognizing that imperative logic is complicated by value judgments, Simon used the term *satisficing* to refer to the fact that the optimal solution is difficult to obtain, and *"figures of merit permit comparison between designs in terms of 'better' and 'worse' but seldom provide a judgment of 'best'"* (p. 138). Neither deductive nor inductive theorizing seems to correspond with Simon's description of optimal solution, but rather a discovery process of trial-and-error searching through declarative space. This search process echoes Weick's sense-making theorizing concept.

In the management field, design thinking has been adopted as a new concept, and it refers to *"the designer's sensibility and methods to match people's needs with what is technologically feasible and what a viable business strategy can convert into customer value and market opportunity."* [20, p. 2]. Martin [21] argues that design thinking relies on abductive reasoning in which sense making of an observation occurs through drawing inference to the best explanation. Peirce [22] argues that *"abduction is, after all, nothing but guessing"* (p. 137) in that its goal is to derive a possible conclusion in terms of what can be possibly true as opposed to declarative logic whose goal is to determine a proposition to be true or false. Abductive reasoning involves drawing a possible precondition from a specific consequence. For instance, one might conclude that (b) failure to incorporate user requirements leads to low user satisfaction from the specific instance that (a) a newly developed system did not lead to high user satisfaction. Such reasoning is considered a fallacy in deductive logic (affirming the consequent), but is acceptable in abduction. Such a conclusion is an acceptable explanation in abduction because (1) it is one of many possible explanations for instance, (2) it is useful in understanding the phenomena, and (3) it can serve as a basis for solving the problem. Comparison of three reasoning approaches is shown in Table 1.

Table 1. Three Theorizing Approaches

	Deductive	**Inductive**	**Abductive**
Purpose	Declarative	Declarative	Post Hoc Ergo Propter Hoc (i.e., "after this, therefore because of this")
Operating Ground	Closed System	Open System	Open System
Logic	Deriving an explanation for a given instance from the existing body of knowledge	Inferring a general conclusion from a specific instance	Inferring satisficing explanation for a specific consequence

The abductive reasoning approach is useful for design theorizing, because the purpose of design theory is to enable search for a *satisficing* solution for a given design problem. Its purpose is not to derive a hypothesis from the existing body of knowledge and test it in a closed system (deductive theorizing); nor does it intend to infer a conclusion from an observation in an open system (inductive theorizing). Consistent with this, Gregor [23] argues that deductive reasoning alone is insufficient in addressing design problems, because for most design problems there exist a range of potential solutions rather than a single standout solution. Deductive and inductive reasoning are certainly applicable and useful for design theorizing, but abductive reasoning may be more important and more common among researchers. Quite possibly, deductive and inductive claims may often be useful as rhetorical vehicles, *post-hoc* rationalizations of messy design theorizing processes, that explain why the design theories that proceed from design science research ought to be accepted as scientifically valid [24].

3 Design Theorizing

There is a substantial body of literature concerning the definition of design theory and what constitutes a design theory. Walls et al [4] define an information system design theory (ISDT) as *"a prescriptive theory which integrates normative and descriptive theories into design paths intended to produce more effective information systems"* (p. 36). In the IS design science research community, design theories are believed to be *prescriptive, practical, basis for action, principles-based, and dualist constructs* [3, 25]. Although design theory is generally believed to be practical, it is argued by many that design theory needs to be grounded in relevant reference theories, e.g., kernel theories [4, 26]. Further, Walls et al [4] elaborate seven components that a design theory should have, including meta-requirements, meta-design, design method, kernel theories, etc. Focusing on the dualistic assumption of design theory, Baskerville and Pries-Heje [25] present a simplified view of design theory that consists of two parts: design practice theory concerning *"the theoretical component about design practice"* and explanatory design theory concerning *"the theoretical component about the design artifact"* (p. 273).

With respect to theorizing for design, Walls et al [4] implicitly discuss how a design theory can emerge by showing the relationships between the components of design theory. Also, Gregor and Jones [3] discuss the relations between different types of theory, implying that theory for design and action can by guided by other types of theory, such as theory for explaining and prediction, theory for predicting, etc. More recently, Gregor [23] presents a high-level framework along with seven principles for design theory development by drawing on distinct characteristics of design science research. While these seminal essays have significantly enhanced our understanding of design theory development, what appears to be missing is a granular understanding of design theorizing process, i.e., what are specific elements and activities involved in the design theorizing process? While we recognize that theory building is a highly creative, thought process that cannot be easily captured in an explicit manner, development of an idealized process for design theorizing can aid both design science researchers and designers in solving so called *wicked design problems* [27].

One reason as to why theorizing for design is not well understood (besides its intuitive and creative nature) may have something to do with lack of consensus on what constitutes a theoretical contribution in design science research. Motivated by this issue, Aier and Fischer [28] present a set of six criteria that can be used to evaluate progress in design theories. Further, Keuchler and Vaishnaive [29] suggest that developing a design theory is inextricably bound to refinement and extension of kernel theories, and what may emerge in this theory refinement process is in fact mid-range theory that is particularly useful for constructing information systems artifact. However, a closer examination of design theorizing process which operates within the human mind is warranted to reveal how theorizing for design actually unfolds. Further, the role that theories play may vary across different design science research projects, i.e., kernel theories, mid-range theories, *post-hoc* rationalizations of design theorizing processes, etc. Although we do not discuss this issue explicitly in this paper, development of an idealized theorizing process may lead to a more differentiated discussion of the necessity of theory and theorizing process in design science research.

4 A Design Theorizing Framework

The need for a focus on design theorizing suggests the potential value of a framework to aid understanding of how we theorize for design and what key activities are involved in developing a design theory. The purpose of the framework is not to prescribe a mechanical method that a researcher can follow to theorize for design, but rather to identify and organize the essential activities in the theorizing process (see Figure 1).

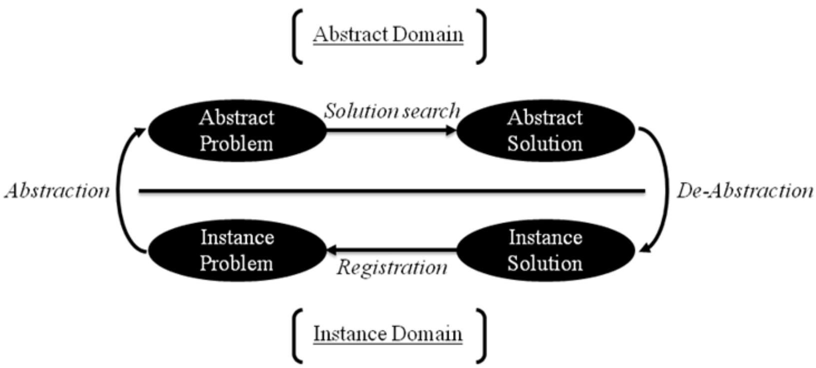


Fig. 1. Design Theorizing Framework

4.1 Theorizing Domains

A key underlying assumption of this framework is that theorizing for design operates in two distinct domains: an abstract domain and an instance domain. In the abstract domain, a solution search process occurs in which an abstract solution is searched for an abstract problem. Simon [19] uses the highway design example to illustrate that solution search process operates at the conceptual level in which specific construction plans, such as particular locations, are not specified. The abstract domain is generalized, operating at a theoretical with a "class of problems", a "class of goals", or a "class of artifacts" [4, p. 42] rather than particulars. In contrast, the instance domain refers to where an instance (particular) solution is applied to address an instance (particular) problem. Further, the two theorizing domains operate on their own independent ground; specifically, there are fewer constraints on the development of abstract solution search process. For instance, the abstract solution is not constrained to or restrained by an instance problem. Following Weick's notion of disciplined imagination, operations within the abstract domain often build on a basis that is explicit, novel, and interesting in a way that *"stands out in* [one's] *attention in contrast to the web of routinely taken-for-granted propositions"* [30, p. 311]. In contrast, an instance domain may not be as novel, and interesting as an abstract domain; most interesting theoretical insights are discovered when researchers think independently from their observations/data [6, 31-32] .

These two theorizing domains highlight the notable duality in design theories. Most pronunciations of design theory elements demonstrate two fundamental parts: a design practice theory and an explanatory design theory [25]. This duality can be drawn from Simon's original work [19] and is very clearly represented in the Walls et al. framing of design theory [4] to include such elements as design methods, meta-requirements, and meta-designs. In the transition from the abstract domain to the instance domain we can locate design practice theory and design methods. Design practice theory mainly concerns bringing a proposed design artifact to life; it emerges by moving from abstract domain to instance domain, i.e., development of an instance solution based on an abstract solution. Design methods operate similarly. In the abstract domain, we can locate explanatory design theory, meta-requirements, and meta-designs. Explanatory design theory concerns *"principles that relate requirements to an incomplete description of an object"* [25, p.273]. Therefore, explanatory design theory emerges out of abstract solution search process where a search for the right set of command variables takes place, and abstract requirements of a design artifact are identified [19]. Walls et al. [4] used the term 'meta-requirements' to show how the functional requirements and basic features that constitute a class of design artifacts are abstract.

4.2 Theorizing Activities

There are four activities in the theorizing framework, each represented by an arrow in Figure 1. These activities are abstraction, solution search, de-abstraction, and registration. Given that all four of these activities may take place as human thought, it may be possible that these occur not cyclically (as represented in Figure 1), or in the order implied by the arrows, but perhaps may arise simultaneously. In terms of activities, please recognize that the framework is an idealization to aid in understanding and comprehending what can be involved in design theorizing. Each activity is described below.

Abstraction. A theory is said to have generalizability when it is applicable across different settings that go beyond a specific setting in which it was tested [33-34]. Generalizability of a theory is a concern to most theorists, as theories that fail to produce generalized inferences are not considered a strong theory, or not a theory at all [5-6, 30]. Design theory is no exception. A strong design theory should show applicability across widely different settings, and address a broad class of design problems. Theory is said to arise from identifying the key links between data and prescriptions (or propositions) by discarding detailed information, and the abstraction is a process of deriving key concepts observed in a specific instance [6]. In design theorizing, abstraction can be realized when a researcher derives common concepts or ideas from an instance problem by removing details pertaining to the context of the instance problem; by doing so, a broad set of problems can be identified. This process of abstraction essentially involves *reflective judgment* where unknown universals for given particulars are sought [35]. When people recognize a problem which cannot be solved intuitively, they rely on their cognitive faculties to distinguish between the peculiarities and the essential conditions for the problem [35-37]. During this process, reflective judgments are called for to understand the problem at a more universal level. Most design problems cannot be solved intuitively or with certainty. It is this uncertain nature of design problems that calls for reflective judgment to decide which essential conditions are applicable to a broader class of problems than just the one at hand. For example, a system user may express his or her frustration for being unable to locate documents effectively in a knowledge management system (an instance problem). Abstraction can be achieved by extracting the key concepts related in the problem, such as user frustration and systems search functionality (an abstract problem).

Solution search. Simon [19] describes how the solution search process (i.e., a goal-seeking system) communicates with the outside environment through two channels: the afferent (the world of the senses) and the efferent (the motor world). The afferent is a sensory world where outside environment is perceived by regarding its state, and the efferent is the world where actions are taken. These two worlds operate at the abstract level; the problem environment is recognized through stored memory information in the human mind, and any particular actions are imaginary [19]. The attainability of goals is determined by making associations between elements of the imagined environment with the elements of the imagined actions. These associations are *"between particular changes in states of the world and particular actions that will bring these changes about"* (p. 141). Thus, theorizing for the solution search concerns understanding relationships between the afferent and efferent, and how the afferent responds to the changes made by actions in the efferent. This process is highly iterative, and requires searching for the right set of actions in the efferent (e.g., creating components of a design artifact) that will bring sufficient changes in the afferent (e.g., solving the requirements of the problem). Thus, an important element in this process is theorizing for the generalized components and the generalized requirements of a design artifact. Each component of the imagined artifacts (the efferent) would have to be theorized individually and collectively in the context of the afferent. The functional explanation of the imagined design artifacts proceeds from this theorizing process, i.e., an explanatory design theory [25].

De-Abstraction. During the solution search, proposed solutions or design artifacts may be theoretical, abstract concepts; these are imagined, generalized problems and solutions. Thus, in order for these to be tested in a specific setting, the generalized, abstract concepts need to be narrowed and instantiated for a particularized setting and a particularized artifact. This de-abstraction involves adding details pertaining to a specific context in which the solution will be applied, and all the details of the instance solution become articulated. De-abstraction essentially requires *deterministic judgment* in which we can subsume given particulars under known universals [35]. De-abstraction is a realization process that may still be partly imaginary; potentially a thought experiment within a design theory is tried as the basis for an imaginary artifact within the designer's mind. Obviously, it may also become partly (or wholly) materialized as an instantiated artifact in reality.

Registration. Whether the design artifact resulting from the de-abstraction process is imaginary or material, the design has to further try this outcome against an instance of the problem setting to verify that the instance outcome has potential to serve the needs of an instance of the problem. Like the de-abstraction outcome, this problem instance may also be wholly imaginary or more-or-less material. Consequently this "trial" may be more-or-less a thought experiment or more-or-less material and empirical. Registering the theory means trying an instance of the solution against an instance of the problem, and adjusting the theory to more exactly correspond to the requirements of the instance. Such adjustments can lead to further abstraction activity as theory adjustments may have more general effects. Because the registration activity is part of the theorizing process, it may be (or may not be) independent of the design science research evaluation process. Evaluation, in design science research, is usually regarded as a validation or proof process where a design theory is shown empirically to stand in terms of how well a resulting artifact performs or to what degree it works as intended [38-40]. While conceptually the evaluation ought to occur after the theorizing process has matured to completion, it could be regarded as a final registration activity in which it is shown empirically that no further adjustment is required.

4.3 Theorizing Threshold

There is no universal starting point in the design theorizing framework. We believe that theorizing begins when some stimulation threshold is exceeded that drives the processes of disciplined imagination and abduction to start one or more of the design theorizing activities. Intuitively, we might think that theorizing always commences with a recognition of an instance problem, and proceeds in the following order; identification of an abstract problem, development of an abstract solution, particularizing an instance of this solution, and registering it to the originating instance problem (this order is indicated by the arrows in the framework). This order would reflect the ideal essence of design science research whose aim is to achieve a clearly stated goal, i.e., bring an intended change to the real world through creation of a new design theory and its resulting artifact [4, 41]. However, reflecting on our own research experience indicates that such an origin for these activities and such an order may be idealizations. It is not the only way that theorizing can take place. As a human sense-making process, theorizing can be messy.

We returned to two of our own design science research projects and reflected on the theorizing process that emerged in these cases. We chose cases that appear in published research to enable interested readers to examine the process and the results of the theorizing more carefully. We selected one case in which the design theorizing threshold was first crossed in the instance domain, and one case in which the design theorizing threshold was crossed in the abstract domain.

Case 1: Crossing the threshold in the instance domain and theorizing the design theory nexus. This work developed a method for constructing decision systems along with instantiations for organizational change decision-making and user involvement decision-making. The theory in this work centered on a conceptual structure called a design theory nexus as a means for addressing the "wicked problem" of multi-criteria decision-making. The instantiations included an IT artifact based on spreadsheet software, used empirically in organizations to help decision makers determine what organizational change approach or what user involvement approach to adopt. A subjective evaluation of the artifacts by participants was positive in terms of their satisfaction in use, and their intention to adopt the outcome artifact results.

In the projects associated with this theory, the theorizing threshold was first crossed in the instance domain, when researchers began conducting search conferences (a form of action research) in an organization to unearth a possible organizational change approach. The results led to a taxonomy of change methods (reported in [42]), which was still very much in the instance domain. Theorizing moved to an abstract domain when the problem was generalized into a form later recognized as multi-criteria decision making (an abstract problem), and the concept of a theory nexus [43] adapted as an abstract solution. The theorizing process returned to the instance domain where it was registered against an organizational change instance. The process went through further abstraction-and-instance cycles that included development of the instance solution for selecting a user involvement approach in IT-developing projects. The theorizing result of these further cycles clarified the aspects of the theory in spanning both single-criterion and multi-criteria decision settings (the user involvement approach proved to be single-criterion - for further case details see [27]).

Case 2: Crossing the threshold in the abstract domain and theorizing a software process improvement design theory. This work developed a design theory that generalized alternative, competing models for improving software organizations. The theory proposed a universal 4-stage model that explained how software process improvement generally progressed in organizations. Each stage was elaborated with a conceptual model of its possible elements. These models were then instantiated with examples (specimens) of published software process and organizational improvement methods such as Six Sigma, CMMI, Balanced Scorecard etceteras. The validation was anchored to the evidence used to instantiate the published examples within the models.

Like case 1, this work developed an initial framework for comparing and contrasting alternative models for improving software organizations. However, this theorizing threshold was crossed initially in the abstract domain as a search for universals in software process improvement. There were no software improvement instances driving this search, just a scholarly curiosity (details of the initial framework are reported in [44]). Unlike case 1, both the abstract and the instances were thought processes; more imaginary than empirical. The instances drawn into the cycles of theorizing were published methods and frameworks. While real to a certain extent, these instances were registered completely through conceptual argumentation rather than field experiments. As the process went through further abstraction-and-instance cycles, the theory was reframed within a body of technological rules as well as the process models (this design theory is elaborated in [45]).

4.4 Discussion

In Case 1, the problem instance and its immediate solution were developed first as a more-or-less un-theorized design, a classification of major organizational change approaches according to their central feature. Such designs have been described as the result of pre-theory in research [46]. In Case 1, the abstract problem and abstract solution were identified later in the research. Our theorizing for Case 1 occurred in the following order: instance problem – instance solution – abstract problem – abstract solution. In contrast, in Case 2 theorizing began with recognition of an abstract problem; that is, a problem was recognized in a researchers' mind at the abstract level (a set of personal/professional experiences may have led to problem recognition at the abstract level, but an instance problem was not yet clearly defined). Our theorizing for Case 2 occurred in the following order: abstract problem – abstract solution – instance solution – instance problem. The fact that theorizing began with identification of an abstract problem independently from any instance problem supports our proposition that theorizing operates in two distinct domains. In Case 2, the search for an abstract solution began without any constraints imposed by an initiating recognition of an instance problem. Importantly in this case, de-abstraction played more significant role than abstraction, as an abstract solution was applied to develop an instance solution and an abstract problem was applied to identify an instance problem.

The two cases are summarized in Table 2. In both cases it is difficult to find any explicit observations that unveil the abstraction process. This is consistent with its nature as a cognitive process of reflective judgment. More detailed research notes would be required; something akin to a personal diary would be necessary to explicate the abstraction activities in these cases. Still these experiences indicate support for the proposition of two distinct theorizing domains, and they show how theorizing can begin at multiple points in the theorizing framework.

Table 2. Examples of Theorizing Thresholds

	Case 1 Organizational Change Nexus	**Case 1 User involvement Nexus**	**Case 2 Software Process Improvement Framework**	**Case 2 Design Theory for Software Process Management**
Abstract problem	Difficulty of choosing among many organizational change approaches	How and when to have user participation in an IT project	What software process improvement approach to use out of many different	Difficulty of designing quality management policies
Abstract solution	Ten generalized change strategies and a way to choose among them	Set of methods and techniques for deciding user participation in IT project	Framework for comparing and contrasting normative models for improving software organizations	Design rules for quality management
Instance problem	Design of organizational change initiatives in two companies	User participation in IT project management in ten companies	Recommending software process improvement approach in a concrete organization	Myriad available quality management fragments

Table 2. (*continued*)

Instance solution	Calculating fit between organizations and change strategies based on query	Applying technological rules in a field study in ten companies	An instantiation strategy of the framework	Design rules based on stages of process improvement models
Abstraction	Problem generalized into a form recognized as multi-criteria decision making	A second iteration realized that this instance could also be abstracted as multi-criteria	Not observed (because this case began with an abstract problem)	From simple rule of thumb in framework to technological rules forming a design theory
Solution search	Realized that it was a form of the concept of a theory nexus	Realized that a theory nexus would be an abstract solution	Identifying 4 universal stages when selecting improvement model	Aligning the technological rules and realizing that the first stage of the four was dominant
De-abstraction	Developed into a spreadsheet and an intervention with management in an organization	Originally developed into specific decision making tool	Made into course material and taught in several professional courses	Made into course material for Graduate & Executive Master level teaching
Registration	Tested in two companies – and later in many	Evaluated with in 10 companies with many project managers	Applied by participants in professional courses	Applied by MPF-participants
Reference	[42]	[27]	[44]	[47]

This empirical evaluation using the past research projects reveals a need for future research. While we believe that the framework proposed in this paper provides a solid conceptualization of design theorizing, further empirical investigation is warranted to critically evaluate and improve the proposed framework. Given that theorizing is a thought process, and can be messy, methods that enable a close examination of human thinking, such as protocol analysis [48-49] and thought experiment [50], may provide a useful means to evaluate and improve the proposed framework. Further, a more systematic analysis of design science research publications is warranted to exploit the different roles that theory play in design science research, and to assess how the proposed framework for theorizing process can be applied and adapted in different cases.

5 Conclusion

In this paper we described and illustrated an idealized theorizing process for design theories. In this process design theorizing operates across two distinct domains; the instance domain encompasses an instantiated solution that is applied to solve an instantiated problem, and the abstract domain encompasses an abstract solution that is devised to solve an abstract problem. We identified four theorizing activities in this process and discussed the role of each activity in developing a design theory. Three conclusions emerge as our contribution.

First, design theorizing necessitates making connections between an abstract domain and an instance domain: abstraction and de-abstraction. Abstraction concerns *reflective judgment* where search for unknown universals for given particulars takes place, whereas de-abstraction concerns *deterministic judgment* where given particulars are subsumed under known universals. Through the process of de-abstraction, design practice theory that involves the instantiation of a proposed design artifact may emerge. Any particular stand-alone design solution that lacks connections with an abstract class of design solutions (and/or a class of design problems) is incomplete as a theory. Particular design solutions may be close to design theories, but the level of abstraction needs to be raised to a class basis that is explicit, novel, and interesting.

Second, a review of the literature on theorizing reveals two dominant theorizing approaches (deductive and inductive) that have been adopted in different research paradigms. While these two theorizing approaches are a useful reasoning tool for theory development, theorizing for design often necessitates adoption of a line of reasoning that is essential for problem solving, i.e., abductive reasoning. Theorizing in design science is abductive because it seeks an imperative logic (rather than declarative) in order to address design problem through the conceptualization of a design artifact. This theorizing process provides a good example of disciplined imagination involving intuitive and creative thinking processes. The adoption of abductive reasoning for design theorizing enables the search for a *satisficing* solution for a given design problem. Further, through the activity of abstract solution search, functional explanations (explanatory design theory) that identification the reasons for meta-requirements result.

Third, reflections on the authors' own prior design science research projects reveal that there is no universal starting point with which design theorizing commences; any origin or ordering in theorizing activities indicated in Figure 1 would be an idealization. A review of two cases shows how the theorizing threshold can be first crossed in either the instance domain or the abstract domain.

Theory is an important and central element in research. Different research paradigms take on different approaches for theorizing. If design theory is a particular kind of theory, it follows that design theorizing may be a particular kind of theorizing. An essential part of design theorizing may involve abductive reasoning because there is a purpose aimed at guiding learning and problem solving. The framework proposed in this paper is aimed toward building strong design theories in the design science paradigm by providing an idealized theorizing process for design science research.

References

1. Winter, R.: Design science research in Europe. European Journal of Information Systems 17, 470–475 (2008)
2. Baskerville, R., Lyytinen, K., Sambamurthy, V., Straub, D.: A response to the design-oriented information systems research memorandum. European Journal of Information Systems 20, 11–15 (2011)
3. Gregor, S., Jones, D.: The anatomy of a design theory. Journal of the Association for Information Systems 8, 312–335 (2007)
4. Walls, J.G., Widmeyer, G.R., El Sawy, O.A.: Building an Information System Design Theory for Vigilant EIS. Information Systems Research 3, 36–59 (1992)
5. Sutton, R.I., Staw, B.M.: What Theory is Not. Administrative Science Quarterly 40, 371–384 (1995)
6. Weick, K.E.: What Theory is Not, Theorizing Is. Administrative Science Quarterly 40, 385–390 (1995)
7. Dubin, R.: Theory Building in Applied Areas. In: Dunnette, M.D. (ed.) Handbook of Industrial and Organizational Psychology. Rand McNally, Chicago (1976)
8. Locke, E.A.: The Case for Inductive Theory Building. Journal of Management 33, 867–890 (2007)
9. Eisenhardt, K.M.: Building Theories from Case Study Research. Academy of Management Review 14, 532–550 (1989)
10. Glaser, B.G., Strauss, A.: The Discovery of Grounded Theory: Strategies for Qualitative Research. Wiedenfeld and Nicholson, London (1967)
11. Popper, K.: The Logic of Scientific Discovery. Unwin Hyman, London (1980)
12. Weick, K.E.: Theory Construction As Disciplined Imagination. Academy of Management. The Academy of Management Review 14, 516–531 (1989)
13. Freese, L.: Formal Theorizing. Annual Review of Sociology 6, 187–212 (1980)
14. Homans, G.C.: Contemporary Theory in Sociology. In: Faris, R.E.L. (ed.) Handbook of Modern Sociology, pp. 951–977. Rand McNally, Chicago (1964)
15. Kaplan, A.: The conduct of inquiry. Chandler, San Francisco (1964)
16. Alvesson, M., Karreman, D.A.N.: Constructing Mystery: Empirical Matters In Theory Development. Academy of Management Review 32, 1265–1281 (2007)
17. Gergen, K.J.: Correspondence versus autonomy in the language of understanding human action. In: Fiske, D.W., Shweder, R.A. (eds.) Metatheory in Social Science, pp. 136–162. University of Chicago Press, Chicago (1986)
18. Henshel, R.L.: Sociology and prediction. The American Sociologist 6, 213–220 (1971)
19. Simon, H.: The Sciences of the Artificial. MIT Press, Cambridge (1996)
20. Brown, T.: Design Thinking. Harvard Business Review, 1–10 (June 2008)
21. Martin, R.: The Design of Business. Harvard Business Press, Boston (2009)
22. Peirce, C.S.: The Logic of Drawing History from Ancient Documents. In: Burks, A.W. (ed.) Collected Papers of Charles Sanders Peirce, Volumes VII and VIII: Science and Philosophy and Reviews, Correspondence and Bibliography, vol. 7. Harvard University Press, Cambridge (1958)
23. Gregor, S.: Building Theory in the Sciences of the Artificial. In: Vaishnavi, V., Purao, S. (eds.) Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology. ACM, New York (2009)
24. Parnas, D., Clements, P.: A rational design process: How and why to fake it. IEEE Transactions on Software Engineering SE 12, 251–257 (1986)

25. Baskerville, R., Pries-Heje, J.: Explanatory Design Theory. Business & Information Systems Engineering 2, 271–282 (2010)
26. Goldkuhl, G.: Design Theories in Information Systems - a need for multi-grounding. Journal of Information Technology Theory and Application 6, 59–72 (2004)
27. Pries-Heje, J., Baskerville, R.: The Design Theory Nexus. MIS Quarterly 32, 731–755 (2008)
28. Aier, S., Fischer, C.: Criteria of progress for information systems design theories. Information Systems and E-Business Management 9, 133–172 (2011)
29. Kuechler, B., Vaishnavi, V.: On the theory development in design science research: Anatomy of a research project. European Journal of Information Systems 17, 489–504 (2008)
30. Davis, M.S.: That's Interesting!: Towards a Phenomenology of Sociology and a Sociology of Phenomenology. Philosophy of the Social Sciences 1, 309–344 (1971)
31. Bacharach, S.B.: Organizational theories. Academy of Management Review 14, 496–515 (1989)
32. Root-Bernstein, R.S.: Discovering: Inventing and Solving Problems at the Frontiers of Scientific Knowledge. Harvard University Press, Cambridge (1989)
33. Lee, A.S., Baskerville, R.L.: Generalizing Generalizability in Information Systems Research. Information Systems Research 14, 221–243 (2003)
34. Yin, R.: Case Study Research: Design and Methods. Sage Publications, Thousand Oaks (1994)
35. Kant, I.: Critique of the Power of Judgment. Cambridge University Press, Cambridge (2000)
36. Dewey, J.: How We Think: A Restatement of the Relation of Reflective Thinking to the Educative Process. Heath, Lexington (1933)
37. Dewey, J.: Logic: The Theory of Inquiry. Holt, Rinehart, & Winston, Troy, MO (1938)
38. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS Quarterly 28, 75–105 (2004)
39. March, S.T., Smith, G.F.: Design and natural science research on information technology. Decision Support Systems 15, 251–266 (1995)
40. Vaishnavi, V.K., Kuechler, W.: Research patterns: Improving and innovating information & communication technology. Auerbach Publications, New York (2007)
41. Gregor, S.: The nature of theory in information systems. MIS Quarterly 30, 611–642 (2006)
42. Pries-Heje, J., Vinter, O.: A Framework for Selecting Change Strategies in IT Organizations. In: Münch, J., Vierimaa, M. (eds.) PROFES 2006. LNCS, vol. 4034, pp. 408–414. Springer, Heidelberg (2006)
43. Carroll, J.M., Kellogg, W.A.: Artifact as theory-nexus: hermeneutics meets theory-based design. In: ACM SIGCHI Bulletin–Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Wings for the Mind, vol. 20, pp. 7–14 (1989)
44. Pries-Heje, J., Baskerville, R.: Improving Software Organizations: An Analysis of Diverse Normative Models. In: Messnarz, R. (ed.) Proceedings of European Software Process Improvement 2003, pp. X.23–X.39. Verlag der Teknischen Universität Graz, Graz (2003)
45. Pries-Heje, J., Baskerville, R.: A design theory for managing software process improvement. In: Vaishanvi, V., Purao, S. (eds.) Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology, pp. 1–2. ACM, New York (2009)
46. Aaen, I.: Essence: facilitating software innovation. European Journal of Information Systems 17, 543–553 (2008)

47. Baskerville, R.L., Pries-Heje, J.: Design and management. In: Simonsen, J., Bærenholdt, J.O., Büscher, M., Scheuer, J.D. (eds.) Design Research: Synergies from Interdisciplinary Perspectives, pp. 63–78. Routledge, London (2010)
48. Isenberg, D.J.: Thinking and Managing: A Verbal Protocol Analysis of Managerial Problem Solving. The Academy of Management Journal 29, 599–775 (1986)
49. Ericsson, K.A., Simon, H.A.: Protocol Analysis: Verbal Reports as Data. The MIT Press, Cambridge (1993)
50. McAllister, J.W.: The evidential significance of thought experiment in science. Studies In History and Philosophy of Science Part A 27, 233–250 (1995)

Forms of Reasoning in the Design Science Research Process

Christian Fischer[1] and Shirley Gregor[2,*]

[1] Institute of Information Management, University of St. Gallen,
Müller-Friedberg-Strasse 8, 9000 St. Gallen, Switzerland
Visitor to The Australian National University, Canberra, Australia ACT 0200
Tel.: +61 2 612 51082
[2] The Australian National University, Canberra, Australia ACT 0200
Tel.: +61 2 61253749
christian.fischer@unisg.ch, shirley.gregor@anu.edu.org

Abstract. Several models for the conduct of design science research (DSR) in information systems (IS) have been suggested. There has, however, been little academic investigation of the basic forms of reasoning underlying these models, namely: deduction, induction and abduction. We argue that a more thorough investigation of these reasoning logics allows for a more comprehensive understanding of the DSR models and the building of information systems design theories (ISDTs). In particular, the question of whether prescriptive design knowledge can be 'theory driven" by descriptive kernel theory can be addressed. First, we show that it is important to distinguish between a context of discovery and a context of justification in theory building and to consider the fundamental forms of reasoning in this light. We present an idealized model of the hypothetico-deductive method, showing how progress is achieved in science. This model includes the contexts of discovery and justification and the matching forms of reasoning. Second, we analyze frameworks for IS DSR and ISDT in comparison with this idealized model. This analysis suggests that few frameworks explicitly refer to the underlying forms of reasoning. Illustrative case studies with first-hand accounts of how IS DSR occurs in practice lend support to the conception of the idealized model. We conclude that work on methodological models for IS DSR and ISDT building would be given a firmer base and some differences in opinion resolved if there was explicit reflection on the underlying contexts of both discovery and justification and the forms of reasoning implicated, as in our idealized model.

Keywords: Information Systems Design Theory, Information Systems Design Science Research, Scientific Method, Design Theory Development, Forms of Reasoning, Abduction, Deduction, Induction.

1 Introduction

A number of frameworks for conducting design science research (DSR) in information systems (IS) have been advanced [1; 2; 3; 4]. This essay concerns the underlying

[*] The authors' names are listed in alphabetical order.

H. Jain, A.P. Sinha, and P. Vitharana (Eds.): DESRIST 2011, LNCS 6629, pp. 17–31, 2011.

logics of the DSR process and how IS design theories (ISDTs) are developed. ISDTs are an output of an IS DSR process and constitute a "general solution to a class of problems" [5, p. 1]. Compared with other theory types in IS [6], they focus on "how to do something", rather than "what is" [7]. As with all theories, however, they are required to be general and therein differ from pure instantiations that solve only a singular problem and not a class of problems [6; 7; 8]. Recognition of ISDTs is important because they are a response to Simon's [9] call for a science of design with knowledge that is at least partly formalizable: that is, theoretical knowledge.

This paper deals with the general questions: "By what means can cognition be achieved?" [10] and how does the development of theoretical knowledge occur in DSR. These related questions have received limited attention in the extant literature. The questions are significant because identifying the forms of reasoning that are implied allows different DSR frameworks to be compared at a fundamental level and also allows comparison with recent work on reasoning logics in the philosophy of science. This examination also allows comparing methods for building ISDTs with methods for building theory in science more generally and examination of how this process is part of the progress of science.

Further, studying DSR methods in terms of modes of reasoning gives insights into issues where there is some debate. For example, there are differing views regarding the relationship between new ISDT and other theory. A distinction can be made between *descriptive knowledge*[1], which is knowledge about naturally occurring phenomena and *prescriptive knowledge* (including ISDTs) which is knowledge of artifacts that are a product of human activity. The feedback loops between descriptive and prescriptive knowledge are what leads to scientific progress and innovation [11; 12]. Some authors propose that DSR can be *theory-driven* [e.g.; 13; 14]. Walls et al. [15] stated that "both the design product and design process aspects of an ISDT **must** be based on kernel theories from the natural or social sciences" [15, p. 48, emphasis added]. On the other hand, it has been argued on logical grounds that prescriptive theory cannot be *derived* from descriptive theory means in any direct way [e.g.; 16].

This paper addresses these issues by showing different ways of achieving cognition—or forms of reasoning—in conducting DSR and contributing to ISDT. Attention to the basic forms of reasoning is important as these forms provide the logical underpinning for any discussion of epistemological approaches. Traditionally, two forms of reasoning have been recognized in philosophy: deductive reasoning and inductive reasoning. A third form of reasoning was recognized more recently by Pierce [17]; specifically abduction.

Our aim in the paper is to show how each of these forms of reasoning is present to varying degrees in IS DSR approaches that have been presented and to demonstrate that the abductive form of reasoning has received insufficient attention. Several case studies of actual design projects are used to illustrate these arguments.

The approach discussed in this paper differs from other authors in the DSR field. Our view rests on the distinction between the context of discovery and the context of justification that can be traced back at least to Reichenbach [18], if not earlier (for an

---

[1] Walls, Widmeyer and El Sawy (1992, p. 41) used the term "kernel theory" to refer to "theories from natural science, social science or mathematics" that are from reference disciplines outside the discipline of information systems. In this essay, the term kernel theory is used interchangeably with descriptive knowledge/theory from all types of science unless otherwise noted.

exhaustive discussion, see [19]). For evaluating the scientific quality of a discovery, only its justification context is relevant (where the scientific quality does not reflect the evaluation of criteria such as utility and novelty). As an illustration, consider the example of the German scientist Kekulé whose discovery of the hexagonal structure of the benzene molecule was inspired by a dream in which he saw a snake trying to bite its own tale [20]. Of course this context of discovery does not comply with any scientific standard. Indeed, Kekulé had to justify his discovery after his dream with scientific methods—and if he had not, his discovery would have never been accepted in science. The context of discovery is rarely considered in the philosophy of science and is often dismissed as being irrelevant for the quality of scientific knowledge. Similarly, the context of discovery has received only limited attention in IS DSR. For example, Goldkuhl [21] focuses on the context of justification by describing ways for grounding ISDTs.

This paper explicitly reflects on the forms of reasoning that can be used in both the context of discovery and the context of justification and it contributes by providing a firmer basis for epistemological discussion of models recommended for IS DSR.

The paper proceeds as follows; first, we describe the basic forms of reasoning employed in scientific research and give an idealized form of the scientific method in terms of the context of discovery and the context of justification. The following section provides a comparison of several IS DSR methods, highlighting the forms of reasoning recognized. The penultimate section provides some examples of reported cases of DSR, with the authors' descriptions of how their research proceeded in practice and the accompanying reasoning steps. Finally, we reflect on our findings and draw implications for approaches used in DSR.

2 Forms of Reasoning and the Scientific Method

2.1 Forms of Reasoning

In this section, we first introduce three basic forms of reasoning: deduction, induction and abduction. Table 1 shows in outline form the nature of each of the types of reasoning, which are discussed further below. In the subsequent section we present the hypothetico-deductive model of science, which includes all three forms of reasoning.

Table 1. Logical Forms *(adapted from Pierce [17, 2.632])*

DEDUCTION Rule. – All the beans in this bag are white. Case. – These beans are from this bag. Result. – These beans are white.
INDUCTION Case. – These beans are from this bag. Result. – These beans are white. Rule. – All the beans from this bag are white.
ABDUCTION Rule. – All the beans from this bag are white. Result. – These beans are white. Case. – These beans are from this bag.

Deductive reasoning is that of a syllogism as discussed by Aristotle in his Prior Analytics [22]. From one or more premises a conclusion can be logically deduced and, usually, the premises are more general propositions than the conclusion. Such a deduction is always certain: that is, if the premises are true, a logically deduced conclusion is necessarily true.

In **inductive reasoning** a general proposition is formulated on the basis of particular propositions. In other words, a sample of entities from a population is observed and these observations are generalized to all entities of the population. Inductive reasoning does not lead to certain propositions, as recognized by Hume [23]. The problem of the uncertainty of inductively gained knowledge has been the starting point for extensive debate in the philosophy of science [cf.; 24]. The problem led Popper to his position of falsifiability: a method employing conjectures and refutations where the emphasis in the scientific model of enquiry was on attempting to falsify hypotheses. The question of how hypotheses arose in the first place was seen as not of interest to science and was to be avoided [25]. Thus, Popper's focus was on the context of justification rather than the context of discovery, a perspective which remains prevalent in many areas today, including behavioral work in the social sciences. For example, a review of research articles in management journals by Colquitt and Zapata-Phelan [26] found that the proportion of theory testing articles was far higher than that of theory building papers, although this trend was declining.

Abduction as a form of reasoning was described by Pierce and is commonly described as an inference to an explanation. Pierce's representation was as follows [17, 5.189]:

> The surprising fact, C, is observed.
> But, if the explanatory hypothesis H was true, C would be a matter of course.
> Hence, there is reason to suspect that H is true.

Pierce characterized abduction as a creative process and noted that abduction is the only logical operation that introduces a new idea [17, 5.171]. Pierce had a broad notion of abduction and his own views changed over time, including abduction as "guessing", as a rational activity and as a heuristic. Later scholars have attempted more precision on the different ways in which abduction can be regarded. For example, Aliseda [27, p. xiii] proposes a general taxonomy for abductive reasoning, which takes into account the nature of the reasoning involved, the type of observation triggering the abduction (novelty or anomaly) and the nature of the explanations produced (facts, rules or theories). It can be seen that abduction as a mode of reasoning is more concerned with the context of discovery rather than the context of justification.

Space constraints preclude a detailed treatment of the differing views of logics of discovery in this essay and we will use the term abduction in the broadest sense to encompass all the different forms of reasoning that are involved in abductive inference, as envisaged by Pierce. Included is creative thinking and what might be termed 'eureka moments', as illustrated in the opening example of Kekulé's molecular insights.

2.2 An Idealized Model for Theory Development

In this section we present an idealized model of a scientific method for theory development to illustrate both the context of discovery and the context of justification and discuss the modes of reasoning that can be employed in each context. The model is recognized as a simplification and we do not wish to claim that there is only one model for scientific progress. Further, Aliseda [27] shows that there is iteration between the contexts of discovery and justification and that in practice it is not easy to separate them as distinct phases. In each context it is likely that all forms of reasoning will be employed to some extent. Here we are focusing on the predominant forms of reasoning likely to be employed in each context.

As a base for our model we use the hypothetico-deductive method, which dates back to Whewell's *A History of the Inductive Sciences* [28]. Figure 1 shows the idealized model with the activities that contribute to theory development in a science where empirical testing of ideas is regarded as a part of scientific activity. Both the context of discovery and the context of justification are recognized.

The process usually begins with a novelty (for instance, a mysterious observation) or with an anomaly. Such an anomaly can be the result of an unsuccessful attempt to test another conjecture. Step 1 in the process, following Pierce [17, 7.218] then involves the abductive form of reasoning:

> *"Abduction, on the other hand, is merely preparatory. It is the first step of scientific reasoning, as induction is the concluding step. [...] Abduction seeks a theory. Induction seeks for facts"* [17, 7.218].

In step 2a, hypotheses are deduced from proposed theories (deduction) and in step 2b, these hypotheses are empirically tested in order to strengthen the reliability of the underlying theory and thereby to generalize the theory (induction) [29].

The first step can also be seen to involve inductive and deductive thinking at times, and, given Pierce's sometimes varying terminology, he could be taken to support this view also. A conjecture could be based by induction upon prior observations [see 30] or partly on deductive thinking where existing theory is seen as applicable in some new setting.

Step 1 describes mainly the context of discovery and steps 2a and 2b relate mainly to the context of justification. However, the process is circular and iterative as steps 2a and 2b might be the starting point for the creation of a new theory, particularly if hypotheses deducted from theory are not supported by observations or if testing reveals a new anomaly, or there may be return to further hypothesis generation, for the same conjecture. All steps taken together lead to theory development.

Some philosophers have emphasized only part of this model. Popper, for example, focused almost entirely on the context of justification and had little interest in step 1—the generation of conjectures. On this point he argued: "The question how it happens that a new idea occurs to a man [...] may be of great interest to empirical psychology; but it is irrelevant to the logical analysis of scientific knowledge" [25, p. 31]. Step 1 is however an important creative task in theory development. Mintzberg [31], referring to Seyle [32], argues that an interesting theory does not stem from a rigorous derivation from existing knowledge, but from generalizing beyond one's data. He concludes with a quotation from Berger: "In science, as in love, a concentration on technique is likely to lead to impotence" [cited after 31; 33].

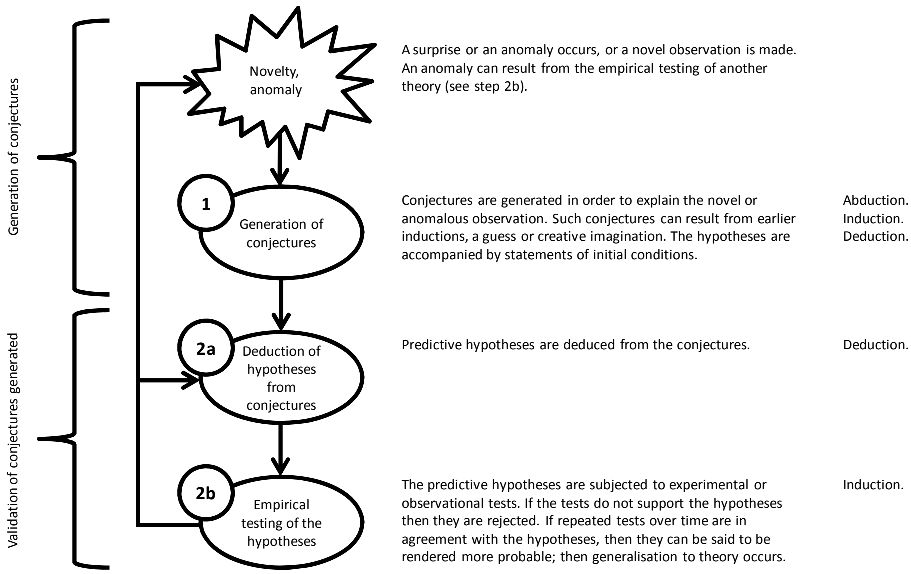


Fig. 1. An Idealized Model for Theory Development Based on the Hypothetico-Deductive Model

3 Forms of Reasoning in DSR Methods

In this section we compare the idealized model of theory development with a number of models for DSR. Justification for this comparison is provided by Bunge, who holds the view that scientific and technological research methods do not significantly differ from each other.

> *"Methodologically, it [technological research] is no different from scientific research. In either case, a research cycle looks schematically like this: (1) spotting the problem; (2) trying to solve the problem with available theoretical or empirical knowledge; (3) if that attempt fails, inventing hypothesis or even whole hypothetico-deductive systems capable of solving the problem; (4) finding a solution to the problem with the help of new conceptual systems; (5) checking the solution, for instance by experiment; (6) making the required corrections in the hypotheses or even in the formulation of the original problem"* [34, p. 174, annotation in brackets added].

The DSR models included in the comparison are those of Nunamaker et al. [2], Hevner et al. [1], Peffers et al. [4] and Takeda et al. [35] (see Table 2). These models were chosen because they have achieved some prominence and have found acceptance as a guide for conducting DSR [36]. Some additional views are provided at the conclusion of this section. Note that Hevner et al. [1] proposed guidelines rather than a process-oriented model. Nevertheless, as Peffers et al. [4] show, parts of their framework can be transformed into the form of a process. The framework developed by Takeda et al. [35] has gained attention recently, particularly in the form adapted by Kuechler and Vaishnavi [3].

A comparison of the DSR frameworks with the idealized form in Figure 1 shows some commonalities as well as some interesting differences. Most DSR frameworks do not explicitly refer to the three basic forms of reasoning, the exception being the framework by Takeda et al. [35].

Three of the four models specify the starting point as a "problem", rather than the novelty or anomaly in the idealized model developed. Nunamaker et al. [2] are the only ones who do not start with a problem, but with a conceptual framework; however, although their conceptual framework gives a short outline of the solution, its purpose is as well to "state a meaningful research question" [2]. We suggest that the difference between the DSR models and the idealized model may be because Herbert Simon's work has been so influential for the DSR models. Simon, with his interest in artificial intelligence techniques and computational logic, focused on well-structured "problems", where the problem-solving process could be seen as means-end analysis (as in backward-chaining techniques). On the other hand, as Aliseda [27] points out, Simon's work focused on one aspect of abductive reasoning, and offers less insight for instances in which genuine novelty or invention occurs: that is, there is no well-structured problem at the beginning. The researcher could imagine both a new opportunity/problem and a potential idea for the solution. The example provided later of Scott-Morton and the first decision support system illustrates this case [37].

Further, there are some differences in the treatment of theory and theorizing, and it is not always clear which steps belong to the development of a design theory and which ones are primarily meant to build a prototype (an "expository instantiation" [7]), for demonstration and evaluation of the design theory. For instance, depending on the particular project, Nunamaker et al.'s [2] step "analyze and design the system" might incorporate the development of theoretical knowledge, the pure implementation of the architecture, or both (cf. Hevner et al. [1] and Peffers et al. [4] analogously). Nunamaker et al. [26] refer to consolidation of experiences learned, which means that some effort is being directed towards inductive derivation of generalized knowledge (theory building). However, few of the DSR models are very explicit about how theory is developed from the research process, inductively or otherwise. In some part this lack of explicitness may be due to differences towards the recognition of design knowledge as "theory" and confusion as to whether it is one of the outputs of DSR. We believe the view expressed by Gregor and Hevner [38] provides a useful way forward, as these authors attempt to harmonize thinking in DSR by showing that DSR contributions can range from less abstract and more specific artifacts or situated instantiations to more abstract and more general emergent design theories about embedded phenomena. This view provides a means of better aligning DSR with other modes of scientific enquiry.

Further, there is evidence of some difference in opinions as to how prior theory, particularly descriptive kernel theory, plays a part in the discovery context and how valid deductive inference is for generation of new design hypotheses. Not shown in Table 2 is the method of Gehlert et al. [13], who propose "theory-driven design research". Their proposal is that descriptive "theory can be used to derive design artifacts" [13, p. 441]. However close reading shows that what they are arguing is that "already existing IS theories **inform** the design researcher when creating IT artifacts"

Table 2. Analysis of Design Science Research Frameworks with respect to an Idealized Research Model

Idealized Research Model [16; 30]	Nunamaker et al. [2]	Hevner et al. [1] (cf. also Peffers et al. [4])	Peffers et al. [4]	Takeda et al. [35], Kuechler & Vaishnavi [3]
Step 0: Novelty/Anomaly	(1) Construct a conceptual framework	(1) Important and relevant problems	(1) Identify problem and motivate, (2) Define objectives of a solution	(1) Awareness of problem
Step 1: Generation of conjectures	(1) Construct a conceptual framework, (2) Develop a system architecture, (3) Analyze and design the system	(2) Iterative search process	(2) Define objectives of a solution, (3) Design and development	(2) Suggestion
Deduction, induction, abduction Creative imagination, guess, observation, prior inductions (i.e. theory), deduction	Deduction, abduction Study of relevant disciplines for new approaches/ideas, develop alternative solutions and choose one solution for investigation	Abduction, induction The knowledge base can be used for artifact development. However, search for satisfactory solutions involves "creation, utilization and assessment of heuristic search strategies" [1, p.89].	Abduction, induction When defining objectives, consider "what is possible and feasible" [4, p. 55]. For design and development, "resources required [...] include knowledge of theory" [4, p. 55].	Abduction "[T]he designer tries to find a candidate [...] [The suggestion subprocess] can be called an abduction process" [35, p. 45].

Table 2 . (*continued*)

Idealized Research Model [16; 30]	Nunamaker et al. [2]	Hevner et al. [1] (cf. also Peffers et al. [4])	Peffers et al. [4]	Takeda et al. [35], Kuechler & Vaishnavi [3]
Step 2a: Generation of hypotheses Formal deduction of testable hypotheses Deduction	(3) Analyze and design the system, (4) Build the prototype system System building is an engineering concept – learn about concepts, frameworks and design through the system building process; gain insight about the problem and its complexity.		(3) Design and development (4) Demonstration In design and development, create the artifact. Demonstrate the use of the artifact, e.g. in experimentation, simulation, case study or proof.	(3) Development "Both the development and evaluation processes are regarded as deduction. [...] [T]he designer applies his/her knowledge to the candidates and obtains what is known at the current state" [35, p. 45]. Deduction
Step 2b: Empirical testing of the hypotheses Strengthen the conjecture by further observation (induction) Induction	(5) Observe and evaluate the system Observation by case study, evaluation by experiment, consolidate experiences learned Induction	(4) Evaluate, (5) Communicate Evaluation criteria: functionality, completeness, consistency, accuracy, performance, reliability, usability, fit with the organization, etc.; methods: observational, analytical, experimental, testing, descriptive. Induction	(5) Evaluation, (6) Communication Observe and measure how well the artifact supports a solution, may include "objective quantitative performance measures […], the results of satisfaction surveys, client feedback or simulations" [4, p. 56]. Induction	(4) Evaluation, (5) Conclusion "[T]he evaluation subprocess uses knowledge to compare those properties obtained in the development subprocess with expectations" [35, p. 45]. Also cf. the remarks above. Induction

[13, p. 442, emphasis added]. We interpret their work to mean that deductive reasoning from prior theory can play some part in the discovery context, congruent with our idealized model, although it can never be a complete guide, with inductive and abductive reasoning still required [16; 31]. Logicians have shown that it is not possible to move from descriptive knowledge to prescriptive knowledge by direct logical inference [see 39]. In fact, if we were to rely on deductive "theory-driven" reasoning in the context of discovery we would not discover anything new, as noted before. As with Kekule's discovery of the structure of the benzene model, a new artifact does have to be justified through testing and through showing, as far as possible, that is it in accord with previously existing knowledge. However, the number of individual design decisions made in the constructions of any reasonably complex artifact, where many alternatives at a decision point may all be in accord with prior knowledge, means that the form that a novel artifact takes will not be completely determinable or explicable by prior descriptive theory.

In conclusion, our analysis supports Bunge's [34] view that scientific and technological research resemble each other on a methodological level, although we have pointed to some interesting differences. Most authors' frameworks for IS DSR or ISDT building do not explicitly mention the three basic forms of reasoning, although they are nevertheless implicitly present in their frameworks, neither do they address how theory development occurs.

4 Illustrative Cases

In this section we provide examples from the literature where researchers report their experiences in DSR and we are able to identify reasoning steps that occurred. Unfortunately there are relatively few examples of cases where a research article describes the sequence of events that led to a final design.

However, some self-reports do occur, even in seminal work on design in IS, that give indications of the thinking processes that occurred. For example, in 1967 Scott-Morton made one of the first steps in the field of decision support systems when he built, implemented and tested an interactive, model-driven decision support tool for his PhD thesis at MIT [37]. His aim was to develop a production planning system with the purpose of helping managers make recurring decision, and he studied outcomes of its use with experiments. Reports show that he began with an interesting idea, that new technology might provide opportunities to give management a new tool, rather than a well-defined problem.

Further first-hand accounts show the use of creativity and imagination in steps of iterative development with DSS artifact construction rather than deductive reasoning from prior theory. Dennis et al. [40] provide a history of the PLEXSYS project at the University of Arizona and list a succession of software tools, initially for automating the systems development process in 1965 and then for group meeting facilities in 1984 and 1987. It is stated that the early stages of thinking about the process of requirements determination depended on "collective wisdom" at the time [40, p. 620], rather than prior theory. Some of this collective wisdom later turned out to be wrong, as in the assumption that the individual or group responsible for the system building project was capable of specifying their requirements, which fact was only recognized through subsequent experience.

Further studies with DSS show hypothesis generation from observation and induction. Grohowski et al. [41] reason inductively and abductively about the use of their electronic meeting systems (EMS) at IBM at 33 sites with over 15,000 people over 3 years to derive some success factors for DSS: for example, "anonymity is particularly beneficial in the meeting process" [41, p. 377], and; "the meeting room environment should match the characteristics of the group" [41, p. 378]. Further design principles that were not originally conceived for the artifact have been extracted from observing the artifact in use: for example, "EMSs help provide an organizational memory concerning related meetings" [41, p. 379], thus giving rise to extended theory inductively.

A further more recent and detailed example is provided by Kuechler and Vaishnavi [3] in their paper on an anatomy of a design project, The focal problem was: "the suboptimal design of business processes due to the lack of incorporation of soft context information into the final designs [3, p. 492]. These authors present their work against the Vaishnavi and Keuchler [42] model of the DSR process. The authors describe an imaginative step, which may be described as their "eureka" moment:

> *"Then, as we reviewed prior approaches to the problem of soft context 'leakage' from system designs we saw that all of them focused on capturing soft context information in some form of graphic notation.* ***Intuitively*** *it seemed that this effort might be misdirected. Based on 20+ years of IS industry development experience we wondered if the real problem was not the capture and representation of soft context information – in most cases the information was available in the original requirements notes – but rather in making that information more immediately available to the designer. Further, as we thought through different soft-information representations of our own, it seemed that a graphic representation of soft or contextual information was the wrong approach. We began to build the position that the highly qualitative, sometimes political, frequently ambiguous nature of soft information was best captured by textual narrative rather than graphics"* [3, p. 493-494, emphasis added].

This extract, which shows the context of discovery, illustrates abductive reasoning in the form of the flash of intuition that led to a conjecture about a problem solution, as well as inductive reasoning from past experience in industry. The authors then describe how:

> *"[W]e began to investigate problem solving cognition and came upon our 'kernel theory' – actually a related set of theories from cognitive, educational, and social psychology that described and explained how varying the presentation of information could enhance or diminish information salience and thus problem solving capabilities"* [3, p. 494].

What can be said here is that prior theory "informed" design: the researchers were able to see that something similar to what was used elsewhere might also hold for modeling notations. This mode of thinking could be regarded as reasoning by analogy, which can be classed as one form of abductive thinking [see 27]. The authors then show how they were able to use suggestions from this prior kernel theory to develop tentative design solutions for their own problem and this reasoning can be seen as in part deductive. In a subsequent development phase the authors show how deductive reasoning from theory and imagination were used to develop a working prototype. This prototype can be taken to encapsulate testable hypotheses (as in, “this prototype is useful in solving the original problem”), which were tested in experiments. This last step represents a justification context and the use of deductive logic.

Thus, this case study shows how induction, abduction and deduction can be used in the context of discovery of DSR and deduction in the justification context. Thus, some support for our depiction of the use of different reasoning modes in the idealized model is provided.

As a final note, it is of interest to consider how the authors in this case summarized their work in the final section of their paper. They state that:

> *“Theories of ‘narrative thinking,’ a mode of cognition receptive to unpatterned information,* ***led to*** *a novel design approach to a conceptual modeling grammar in the suggestion phase”* [3, p. 499, emphasis added].

The authors have emphasized the use of prior theory, rather their insights and imaginative thinking, as the source of their design solution. We may be reading more into the authors’ choice of words than they intended, but we take this statement as showing that deductive reasoning is privileged above abductive and inductive thinking in research reports, even if this emphasis is perhaps unconscious.

5 Conclusion

In this paper we first discussed the role of the three basic forms of reasoning and developed an idealized form of the scientific process based on the hypothetico-deductive model, which includes both a context of discovery and a context of justification. We then analyzed frameworks for IS DSR and ISDT building by comparison with the idealized model and presented some illustrative examples of DSR processes.

We conclude from our analysis that, congruent with Bunge [7], DSR methodologically is not in essence different from an idealized model of the scientific method. This conclusion is important as it adds weight to Simon’s belief that a science of design can be made “intellectually tough” [9, p. 112].

Comparison of the idealized model with current DSR frameworks, however, showed that few have paid explicit attention to the modes of reasoning that are involved. There is a lack of agreement on what occurs in the context of discovery, with little or no recognition in any single framework that the first stage of DSR can involve **all** of abductive, inductive and deductive thinking. In fact, there is some tendency to revert to the Popperian preference for deductive thinking, with claims that design can be ‘theory-driven’ by descriptive theory [13; 14; 15; 43]. Our argument is that these claims are logically indefensible with the design of artifacts, especially novel artifacts. It is important to pay greater attention to the place of abductive reasoning in design research, with our illustrative examples showing how abductive, inductive and deductive reasoning occurs in the context of discovery – the artifact design and development stages.

Further, the existing DSR frameworks do not place much emphasis on inductive reasoning in the building of generalized abstract knowledge (theory), although the illustrative examples of DSR show that general prescriptive propositions are advanced inductively (e.g., see [41]).

We argue that methodological frameworks for IS DSR and ISDT building would be improved and could be better integrated if the three basic forms of reasoning and the idealized model of scientific enquiry were considered more explicitly. More specifically, it should be realized that abduction, deduction and induction are all valid modes of reasoning in artifact development and that induction should be recognized for its role in developing generalized abstract knowledge and theory.

References

1. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28(1), 75–105 (2004)
2. Nunamaker, J.F., Chen, M., Purdin, T.D.M.: Systems Development in Information Systems Research. Journal of Management Information Systems 7(3), 89–106 (1991)
3. Kuechler, B., Vaishnavi, V.K.: On Theory Development in Design Science Research: Anatomy of a Research Project. European Journal of Information Systems 17(5), 489–504 (2008)
4. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems 24(3), 45–77 (2007)
5. Baskerville, R.L., Pries-Heje, J., Venable, J.: Soft Design Science Methodology. In: 4th International Conference on Design Science Research in Information Systems and Technology (DESRIST 2009), pp. 1–11. ACM, New York (2009)
6. Gregor, S.: The Nature of Theory in Information Systems. MIS Quarterly 30(3), 611–642 (2006)
7. Gregor, S., Jones, D.: The Anatomy of a Design Theory. Journal of the Association for Information Systems 8(5), 312–335 (2007)
8. Vahidov, R.: Design Researcher's IS Artifact – A Representational Framework. In: Proceedings of the First International Conference on Design Science Research in Information Systems and Technology (DESRIST 2006), pp. 19–33 (2006)
9. Simon, H.A.: The Sciences of the Artificial. MIT Press, Cambridge (1996)
10. Becker, J., Niehaves, B.: Epistemological Perspectives on IS Research: A Framework for Analyzing and Systematizing Epistemological Assumptions. Information Systems Journal 17, 197–214 (2007)
11. Mokyr, J.: The Gifts of Athena: Historical Origins of the Knowledge Economy. Princeton University Press, Princeton (2002)
12. Stokes, D.: Pasteur's Quadrant: Basic Science and Technological Innovation. Brookings Institution Press, Washington (1997)
13. Gehlert, A., Schermann, M., Pohl, K., Krcmar, H.: Towards a Research Method for Theory Driven Design Research. In: Hansen, H.R., Karagiannis, D., Fill, H.G. (eds.) Proceedings of the 9th International Conference Wirtschaftsinformatik, Business Services: Konzepte, Technologien, Anwendungen, Österreichische Computer Gesellschaft, Wien, pp. 441–450 (2009)
14. Arazy, O., Kumar, N., Shapira, B.: A Theory-Driven Design Framework for Social Recommender Systems. Journal of the Association for Information Systems 11(9) (2010)

15. Walls, J.G., Widmeyer, G.R., El Sawy, O.A.: Building an Information System Design Theory for Vigilant EIS. Information Systems Research 3(1), 36–59 (1992)
16. Gregor, S.: Building Theory in the Sciences of the Artificial. In: Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology. ACM, New York (2009)
17. Pierce, C.S.: Collected Papers of Charles Sanders Peirce. In: Collected Papers of Charles Sanders Peirce, vol. 1-6 edited by Hartshorne, C., Weiss, P., vol. 7-8 edited by Burks, A.W. Harvard University Press, Cambridge, MA (1931-1958)
18. Reichenbach, H.: Experience and Prediction: An Analysis of the Foundations and the Structure of Knowledge. University of Chicago Press, Chicago (1935)
19. Hoyningen-Huene, P.: Context of Discovery and Context of Justification. Studies in History and Philosophy of Science. Part A 18(4), 501–515 (1987)
20. Rothenberg, A.: Creative Cognitive Processes in Kekulé's Discovery of the Structure of the Benzene Molecule. The American Journal of Psychology 108(3), 419–438 (1995)
21. Goldkuhl, G.: Design Theories in Information Systems – A Need for Multi-Grounding. Journal of Information Technology Theory and Application 6(2), 59–72 (2004)
22. Smith, R. (ed.): Aristotle: Prior Analytics. Hacket Pub., Indianapolis (1989)
23. Hume, D.: A Treatise of Human Nature. In: Bigge, L.A.S. (ed.), Clarendon, Oxford (1988)
24. Vickers, J.: The problem of Induction. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy (2010)
25. Popper, K.R.: The Logic of Scientific Discovery. Hutchison, London (1959)
26. Colquitt, J.A., Zapata-Phelan, C.P.: Trends in Theory Building and Theory Testing: A Five-Decade Study of the Academy of Management Journal. Academy of Management Journal 50(6), 1281–1303 (2007)
27. Aliseda, A.: Abductive Reasoning. In: Logical Investigations into Discovery and Explanation. Springer, Dordrecht (2006)
28. Whewell, W.: A History of the Inductive Sciences (Parts I, II, II). In: Nelson, T. (ed.) (1967)
29. Burch, R.: Charles Sanders Peirce. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy (2010)
30. Butts, R.: Hypothetico-Deductive Method. In: Audi, R. (ed.) The Cambridge Dictionary of Philosophy. Cambridge University Press, Cambridge (1999)
31. Mintzberg, H.: Developing Theory about the Development of Theory. In: Smith, K.G., Hitt, M.A. (eds.) Great Minds in Management. The Process of Theory Development, Oxford, pp. 355–372 (2005)
32. Seyle, H.: From Dream to Discovery. McGraw-Hill, New York (1964)
33. Berger, P.L.: Invitation to Sociology: A Humanistic Perspective. Penguin Books (1967)
34. Bunge, M.A.: Philosophical Inputs and Outputs of Technology. In: Bugliarello, G., Doner, D.B. (eds.) The History of Philosophy and Technology. University of Illinois Press, Urbana (1979)
35. Takeda, H., Veerkamp, P., Tomiyama, T., Yoshikawa, H.: Modeling Design Processes. AI Magazine 11(4), 37–48 (1990)
36. Hevner, A.R., Chatterjee, S.: Design Research in Information Systems: Theory and Practice. Springer, Heidelberg (2010)
37. Scott-Morton, M.: Computer-Driven Visual Display Devices – Their Impact on the Management Decision-Making Process. Harvard Business School, Boston (1967)
38. Gregor, S., Hevner, A.R.: Introduction to the Special Issue on Design Science. Information Systems and E-Business Management 9(1), 1–9 (2011)
39. Edgley, R.: Reason in Theory and Practice. Hutchison University Library, London (1969)

40. Dennis, A.R., George, J.F., Jessup, L.M., Nunamaker Jr., J.F., Vogel, D.R.: Information Technology to Support Electronic Meetings. MIS Quarterly 12(4), 591–624 (1988)
41. Grohowski, R., McGoff, C., Vogel, D., Martz, B., Nunamaker, J.: Implementing Electronic Meeting Systems at IBM: Lessons Learned and Success Factors. MIS Quarterly 14(4), 369–383 (1990)
42. Vaishnavi, V., Kuechler, W.: Design Research in Information Systems (2004), `http://ais.affiniscape.com/displaycommon.cfm?an=1&subarticlenbr=279` (last updated August 16, 2009, last accessed March 2, 2011)
43. Dietz, J.L.G.: Enterprise Ontology – Theory and Methodology. Springer, Heidelberg (2006)

Enhancing Design Science through Empirical Knowledge: Framework and Application

Janusch Patas, Danijel Milicevic, and Matthias Goeken

Frankfurt School of Finance & Management, IT-Governance-Practice-Network,
Sonnemannstraße 9-11, 60314 Frankfurt am Main, Germany
{j.patas,d.milicevic,m.goeken}@fs.de

Abstract. The discourse about differences between behavioral and design science still attains wide interest in the information systems research community. While design-oriented research is repeatedly subject to criticism on account of lacking transparency and rigor, behavioral research is fighting against the accusation of little relevance. It would be highly desirable to overcome the shortcomings of design science by using existing theories, empirical knowledge, etc. within the design of an artifact. For that purpose, we present a framework that shows how different ways of applying empirical knowledge can put the research of design scientists on a better grounding and thus improve the rigor of design science. Specifically we point out, how design science can be performed more rigorously on the basis of our framework by empirically motivating, guiding, evaluating, and analyzing design science research. To illustrate the application of our framework, we will provide an example from the domain of information security.

Keywords: Design Theory, Empirical Knowledge, Framework.

1 Introduction

Regionally bound research traditions, diverse intellectual background, differences within the education of scholars of the field, etc. [1] are responsible for the different types of research results found in information systems research (ISR). Consequently various types of research results are generated due to different research paradigms, which are distinguished into a design-oriented (design science) and a behavioral (behavioral science) paradigm [2]. Design science research in information systems (DSRIS) is mainly concerned with the construction of IT artifacts[1] that are intended to solve relevant problems in an innovative way or optimize existing solutions for relevant problems and therefore belongs to design science [3]. Business & information systems engineering (BISE), as prevailing within the German-speaking domain is on of the biggest communities of design scientists [4]. In contrast, behavioral ISR, predominately located within the Anglo-American domain, deals with the development and justification of theories which explain and predict phenomena related to the use of information systems [2].

[1] IT artifacts are constructs, models, methods, and instances [2, 10].

H. Jain, A.P. Sinha, and P. Vitharana (Eds.): DESRIST 2011, LNCS 6629, pp. 32–46, 2011.

Because of that, varying types of research results can not only be traced back to the mentioned regionally bound research traditions, but also to different Denkstile (rooted in engineering vs. natural-science-oriented research approaches), which determine the application of diverse research methods or the respective research goals (truth vs. utility). Synergy effects and further potential, which could result from overcoming the borders of these paradigms, are hardly being realized [5]. But, some authors propose and call for a mutual exchange between design-oriented and behavioral research [1, 6]. This is regarded as being useful in the following ways: For one, empirical research results can trigger design-oriented research by offering insights into and pointing to interesting and relevant phenomena. Understanding the cause of and having the knowledge about problems related to studied phenomena could motivate design-oriented researchers to solve these problems with the means of innovative or improved IT artifacts [7, 8, 9].

Following this line of thought, decisions within the design process of IT artifacts could be more strongly linked to and grounded in theoretical knowledge or empirical evidence to account for more transparency and rigor within DSRIS [2, 4]. An alternative perspective is viewing IT artifacts as subjects of empirical research, e.g., in behavioral science. On the one hand, this would allow for a broader evaluation of the utility of the IT artifact [2, 10]. On the other hand, it would foster the production, evaluation and use of knowledge about the IT artifact [6].

In this paper we propose a framework to support a stronger collaboration or even integration of empirical research into DSRIS from a design science perspective. In doing so, we address the different ways of leveraging the benefits mentioned above. As our main contribution we distinguish between different types of application of empirical knowledge in design science and we outline how design science research can be improved by additional rigor in certain parts of the research process.

The remainder of this paper is structured as follows: First we will establish an essential terminology to fall back upon in the subsequent sections and to increase the transparency of our framework construction. In section 3.1 we will derive all important elements of our framework to explain their interplay in section 3.2. In section 4 we demonstrate the application of our framework with the aid of an example from information security research. Section 5 discusses our findings and in section 6 we provide concluding remarks and describe potential future research.

2 Basic Considerations and Related Work

In order to better understand the divide as well as potential points for collaboration between the two paradigms of research, it seems helpful to go back to the proverbial roots. By shedding the labels for a moment we strive for a new perspective on the paradigms and their divide. In this section we conceptualize ways of leveraging the synergy and potential between the two research paradigms by broadening our view to two of Aristotle's virtues of thought: episteme and techné. Additionally we derive what we consider approaches of design science which help us distinguishing between the scientific process in design science and design practice.

A first step is establishing what science and thus the scientific process is. In science we search for truth. The Greeks philosophers described the transition of *that* *which was believed to be true* (doxa) to *that which was known to be true* (episteme) as the scientific process [11]. Doxa and episteme can be considered as the two extremes of this spectrum of certainty in knowledge regarding its truth. Along with this dimension of truth, there is also the dichotomy of theory and practice. The Greek philosophers, especially Aristotle in his work Nicomachean Ethics [12], talk about episteme and techné. Episteme was considered scientific knowledge with the connotation of absolute certainty, which differs from our contemporary understanding and use of the term. An example used by Aristotle was the field of mathematics, especially geometry, where formal proofs can lead to absolute certainty about knowledge ("We are all convinced that what we know through episteme cannot be otherwise than it is", [12]). Such certainty is hard to find in nature and the question of how we can know something to be true led to a refined definition of knowledge, in which knowledge can only be asserted. By means of experimentation and (empirical) evidence, the knowledge we attempt to gain is "conceived in a probabilistic sense" [11]. By deploying the scientific process (i.e. moving further away from doxa, the mere belief in knowledge), we develop knowledge with a higher certainty of truth. In our contemporary understanding episteme are theories, which were hypotheses (doxa) until – through scientific methods – our certainty in the truth of this knowledge rose.

Another virtue of thought is techné, which describes practice and is often being translated as craft or art [13]. The actions of a practitioner are guided by another type of knowledge. Philosophy knows the term practical knowledge, which according to Aristotle – as a result of practical thinking – can also be true or falsify, but only in relation to the desire of the practice or action. A contemporary interpretation of the term desire in this context would be purpose. As such practice also holds a form of knowledge; however its truth or falsification can only be determined in relation to the purpose the practice had. This practical thinking however does not derive from theoretical thinking, but from means like trial & error. What deviates from episteme is the lack of a scientific process about techné in the writings of the Greek philosophers. The systematization of such knowledge and treatment with scientific methods is a research endeavor in the field of design science research for some time now. As such "…techné expresses a demand for a theoretical awareness which, so to speak, justifies conceptually that practical knowledge which is already established empirically. Techné consolidates this practical knowledge and affords it a certain extension – due to the inherent generality of theoretical knowledge – …" [14].

This consolidation has made major progress in the past. We explore opportunities from the design scientists' point of view by learning from and leaning on inform design science research. In order to do so the duality of design needs to be discussed. The word design can be both a noun and a verb (or i.e. a product or a process) [2]. Design as a noun is the proposition that a certain artifact (the result of design science), as a combination of its components, can generate the desired utility (serve its purpose), which is essentially knowledge in the uncertain state of doxa. Especially when developing an innovative artifact the outcome of a proposed design is not absolutely clear. Once the artifact is built, it can be evaluated in regard to whether or not it serves its purpose. Confirmation of the artifact doing so (e.g. by an empirical study) is a step towards a higher level of certainty about the initial proposition. Such explanations have been associated with terms like "design theory" or "explanatory design theory" [15]. This is one of the approaches in design science and is similar to the process of theory building with a delayed (due to the construction of the artifact) theory testing known from sciences which are assigned to the virtue episteme.

The other aspect is design as a verb or process. According to [15] a design practice theory would describe how the designer achieves his design goal and builds an artifact; an action also known as projection [16]. As such a potential design practice theory would structure and reason the projection of functional explanations from an explanatory design theory to the final artifact. However, the possibility of a design practice theory is part of an on-going discussion in the ISR community [17, 4].

Design science, as the "science of the artificial" [18] has the artifact at its core. From motivating the need for an artifact to its construction and concluding evaluation, the majority of activities in design science [2, 19] are centered on it. [10] explicate that behavioral science – as a form of natural science – investigates both natural and artificial phenomena. It seems helpful to use this distinction in regard to a science of the artificial in order to find opportunities to use or generate episteme and thus integrate both research paradigms tangibly. We will subsume sciences that investigate natural phenomena as reference theories about natural phenomena [20], indicating that ISR as a science discipline lends from the rich set of theories from other sciences, in absence of ISR kernel theories [10, 2]. Sciences that are linked to episteme (and as such generate scientific knowledge), but theorize about artificial phenomena are labeled reference theories about artificial phenomena. We consider this a third approach to theorizing design science research (along with explanatory design theory and design practice theory), which seems to be a recent development [20, 6].

In this section we showed the general distinction between episteme and techné, or their contemporary counterparts: theory and practice. We identified three approaches within design science research which attempt to theorize design science and help distinguish the scientific aspect from design practice. Building on this we will show means of motivating, guiding, evaluating and analyzing design science research with the help of scientific knowledge from the behavioral science in ISR and other sciences in order to improve the rigor of design science and help with a theoretical foundation of design science research activities.

3 Framework for Empirical Based Artifact Design

3.1 Essential Elements of the Framework

As a preliminary step, we derive four types of support for design science from the two so called 'useful ways of mutual exchange' mentioned in the introduction. The four types furthermore rest upon [2, 19]: Table 1 shows the relation between these four types of empirical support and the design science process model by [19]. This is done to convey how the research model integrates with the construction process of an artifact to increase the rigor of said process. Additionally the design science guidelines by [2] are linked to show where our framework makes a contribution to design science in general.

Table 1. Types of Application of Empirical Knowledge and their Relatation to [2, 19]

Types of Application of Empirical Knowledge		DS Process Model [19]	DS Guidelines [2]
1	Establish DS problems and motivate design empirically	Problem identification and motivation (activity 1)	Problem relevance (Guideline 2)
2	Support design decisions and design actions empirically	Define the objectives for a solution (activity 2); Design and development (activity 3)	Research rigor (Guideline 5)
3	Evaluate design result empirically	Evaluation (activity 5)	Design evaluation (Guideline 3)
4	Analyze design result empirically	n/a	Research contribution (Guideline 4)

Reference Theories of Natural Phenomena

The objective attributed to reference theories of natural phenomena is to „… explain how and why things are" [10]. Since these disciplines are developing insights into natural phenomena they consequentially do not generate knowledge about artifacts. From a design science perspective this kind of research can be labeled also non-artifact-centric research. The outputs of these research endeavors are empirically tested hypotheses, which ideally come attached with certain empirical evidence. Such insights into natural phenomena are of importance for design science, as design science researchers try to influence these by constructing an artificial object [21, 10] to e.g., emulate the cause of a desired phenomenon. From the design scientists' perspective, they can be regarded as reference theories [9, 10, 20] which explain and make predictions about natural phenomena [22]. Such explanatory and predictive theories are based on causes and effects [23].

The element 'cause' is in a directional relationship with the element 'effect', with both being parts of the element hypotheses (figure 1). From empirical testing of such hypotheses we gain a level of empirical evidence [24], which we earlier referred to as the level of certainty about knowledge to be true.

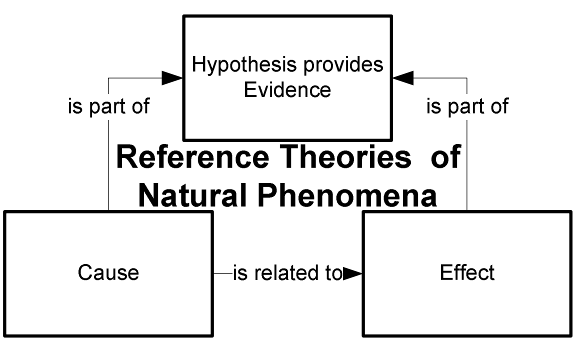


Fig. 1. Elements from Theories of Natural Phenomena

From a design science perspective this evidence about explanations or predictions of natural phenomena can be used to generate or improve requirements for the design, which allows grounding of the problem statement with empirical data [7, 8, 9, 18]. Furthermore this knowledge can be the motivator for the construction of the artifact to solve an existing problem (e.g., to address the lack of a cause for a desired effect).

Explanatory Design Theory

Design theories can be operationalized for the construction of an artifact [17]. In their work, [15] separate the theoretical component about design practice from the theoretical component about the design artifact. This is in accordance with the aforementioned duality of the design term. The explanatory design theory holds functional explanations about „…, why designs and artifacts have certain attributes and features“ [15] and calls for the decomposition of the design goal into concrete requirements, which have to be fulfilled by the components of the artifact (Figure 2). The design goal can be based on reference theories in order to motivate and reason the construction of the artifact based on empirical research.

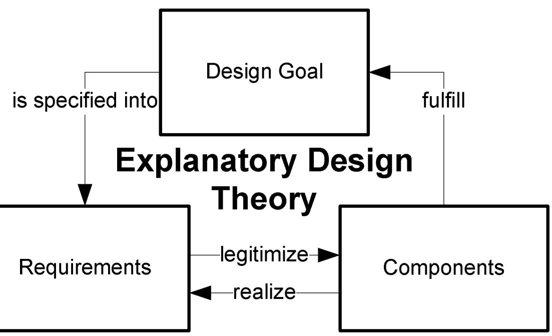


Fig. 2. Explanatory Design Theory according to [15]

Components are in a functional relationship with their requirements. „ The nature of the requirements explains the incomplete description in terms of the requirements.“ Additionally the components are being legitimated by the defined requirements [15]. The requirements derived from the design goal can be considered the ends of the design and the components the means. This way a means-ends-relation can be derived [7]. The explanatory design theory should ensure the internal cohesion and consistency of an artifact [25]. It can be interpreted as a blueprint which describes the requirements and components needed to fulfill these, but does not instantiate these component types.

Design Practice Theory

A design practice theory describes the practical way of „…how to design something“[15]. [23, 26] refers to theories for design and action and describes prescriptive technological rules, which state: „… if you want to achieve Y in situation Z, then perform action X“ [5]. This prescriptive procedural knowledge can for example be structured by a design method [9]. According to [27] such a design method is an (IT) artifact itself, which takes the form of a meta-design that needs to be instantiated for specific situations. To consider this aspect and avoid a too narrow interpretation of technological rule as a method, we're using the term design action (Figure 3).

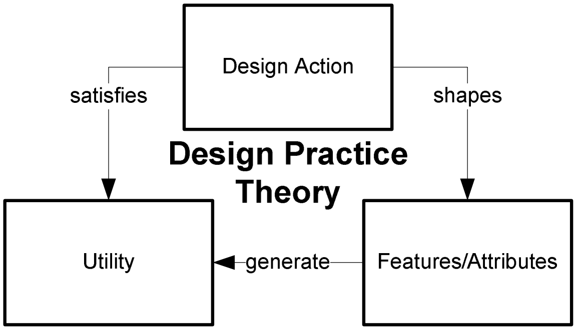


Fig. 3. Design Practice Theory

Design action configures the features and attributes of the components, which were defined earlier in the explanatory design theory. These configured components then generate utility (regarding the task of fulfilling the requirements). The element design action can be guided by episteme (in form of theoretical or empirical knowledge) in the configuration process of the components. By doing so, the design decisions and actions can be justified with empirical data which makes the design practice more rigorous as it becomes more transparent and well-reasoned.

Reference Theories of Artificial Phenomena

Along with natural phenomena also artificial phenomena exist [10, 18]. In principle research streams that focus on artificial phenomena follow the same processes as research does which theorizes about natural phenomena. As described in section 2 this kind of research deviates slightly. [20] describes it as additional features), because between doxa and episteme there is the construction of the artifact, which is the subject of the research. The aspect which [20] labels as "exterior mode of design disciplines which theorize about artifacts in use" is found also in [6] in the form of so called design knowledge. To emphasize the focus on the artificial in this kind of research, we subsume it under the label artifact-centric research.

In artifact-centric research the knowledge about an artifact (or artificial phenomenon) is gained by the use of scientific method to identify context-specific effects and side-effects of the artifact use or costs that arise from using the artifact for its intended purpose [6]. Artifacts that show similar means-ends-relations can be tested using comparative requirements [6], e.g. to determine what artifact generates highest utility in regard to the chosen requirements. Such a comparison can be based on various sets of features and attributes [20]. In contrast, the evaluation of the artifact is assigned to the design process of the artifact and determines the utility based on specific criteria [2, 19]. The artifact-centric research develops insights into features or attributes of the artifact, which can be transformed into technological rules, which then can support design decisions and design action [20]. Knowledge about the artifact is generated and the artifact is analyzed for both intended and unintended effects of its. Additionally such research can motivate the design of a new artifact or an improvement of an existing artifact based on its defects.

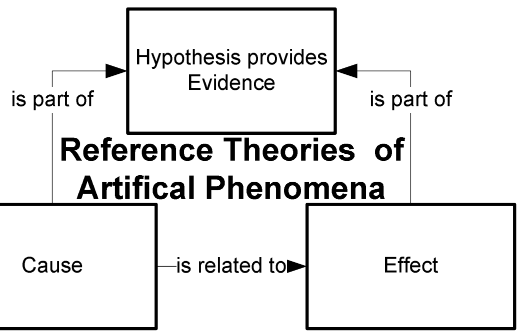


Fig. 4. Elements of Reference Theories of Artificial Phenomena

Figure 4 illustrates the elements and the similarity to the cause and effect relation found in reference theories of natural phenomena. However we substitute – or rather specify – the cause for the artificial phenomenon with the artifact to also show the subtle differences. We propose that theories about artifacts can be one research output of such endeavor.

3.2 Framework of Empirically Supported Design Science

In this section we will merge the quadrants, the four types of theories described above and their elements together what will result in our framework (figure 5). The framework is divided into two columns representing on the left side episteme and on the right side techné. In the upper left quadrant knowledge concerning natural phenomena is generated, which can be found consolidated as reference theories of natural phenomena. Knowledge concerning artificial phenomena, which is or can be found consolidated as reference theories of artificial phenomena, is located in the lower left quadrant. As show in figure 5 non-artifact-centric knowledge as well as artifact-centric knowledge is handed over to the entire right techné column. It is illustrated by the two depicted arrows, named non-artifact-centric and artifact-centric knowledge, as each of these arrows touches the entire column. In the upper right quadrant of the techné column the explanatory design theory is located which holds the functional explanation of the designed IT artifact. Straight under the explanatory design theory, the design practice theory has its position. The design practice theory supports shaping the attributes and features of the above defined components.

The resulting IT artifact, which is the final design product, has to be handed over from the techné column to the lower left quadrant in the episteme column. This particular transfer is restricted to the lower left quadrant as only there is knowledge generated concerning artificial phenomena. This is indicated by the arrow named artifact that touches solely the quadrant in the lower left corner.

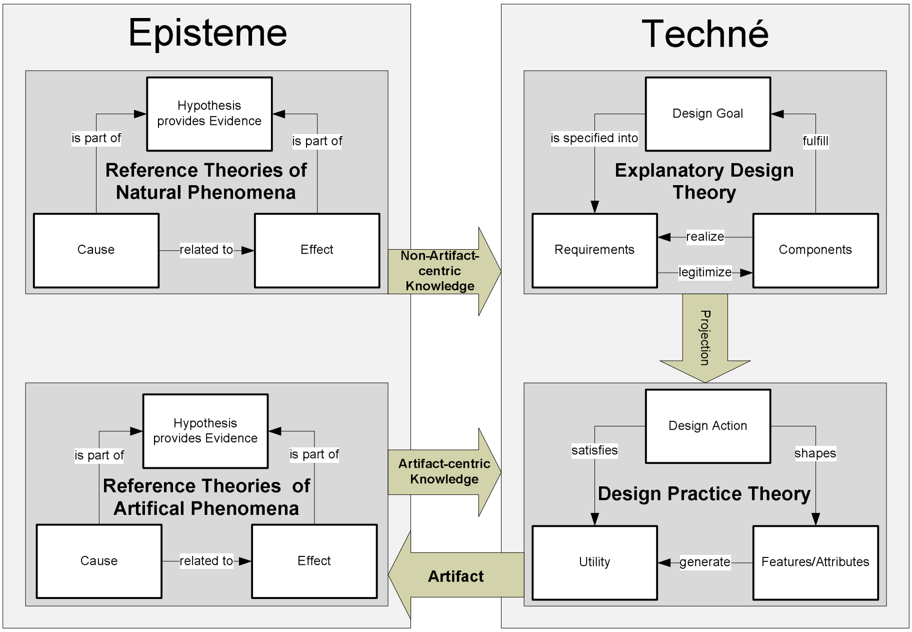


Fig. 5. Research Framework for Empirical-Driven Design Science (EDDS)

The following part of this section is reserved to explain the integration of empirical knowledge into design science. Therefore, we callback to our four types of application of empirical knowledge from table 1; these are explained by the means of our framework. Establishing the problem and motivating the design of an IT artifact can be supported by non-artifact-centric as well as by artifact-centric knowledge. In the first case, theoretical or empirical knowledge is taken from a reference discipline. This knowledge serves to establish the problem more rigorously and to motivate the design of an IT artifact solving the investigated problem. In the second case, an unintended side effect is discovered within the artifact-centric research that is caused by an IT artifact, for instance. Thus, this can also serve to establish and motivate a problem to be solved.

Both design decisions and actions can be supported by non-artifact-centric and artifact-centric knowledge stemming from the episteme column. Whereas design decisions are made at the level of the upper right corner, as the structure and the interplay of the components are defined related to the requirements derived from the design goal, within the design practice theory design actions are responsible shaping the features and attributes which are instances of the components. Artifact-centric and non-artifact-centric knowledge can be utilized for the definition of the requirements regarding the components. Similar holds true for the design practice theory, but in contrast from artifact-centric knowledge very concrete technological rules can be derived [20]. Technological rules can also be derived from non-artifact-centric knowledge. Anyway, the derivations of these technological rules have left some room for interpretation [5].

The evaluation of the design result, the IT artifact, is an intrinsic function of design science and therefore, takes place right after the design and development of the artifact. Noticeably, as mentioned before, artifact-centric research has to be distinguished from artifact evaluation, because it differing aims reach beyond an evaluation. Regarding the evaluation issue, there is a plethora of literature available in DSISR. Hence, we do not stress this topic in this paper and thus refer to relevant literature [2, 28, etc.]. Analyzing the design results is in the scope of artifact-centric research. The IT artifact resulting from the design practice can be transferred to the lower left quadrant as illustrated by an arrow in our framework in figure 5. Herein, new knowledge is generated about the artifact and from there about artificial phenomena. New and innovative knowledge about the artifact and its application domains can be analyzed whereby new knowledge and also new technological rules can be passed to the techné column.

4 Application of the Framework

To further illustrate the application of the framework we will give a brief example of how the framework can be applied to design science research which draws from empirical knowledge. This example from the information security domain will describe the design and development of a password policy. The aim is to collect empirical knowledge relevant to the design artifact and deduce from empirical findings (cause-and-effect relationships) guidance on 1) what needs to be considered (requirements), 2) how the artifact should be constructed (selection and configuration of components) and 3) what effects are to be expected from the application of the artifact (utility and side effects).

Table 2 lists empirical knowledge that is found in the left column of the framework (episteme). The last column shows where it can be applied in the design science process model suggested by [19]. Terms from our framework are underlined in the table. The table presents a small selection of research that may be relevant for the desired design artifact. Even with such a small set of empirical studies the importance of cause-and-effect-like relationships becomes clear to the design scientist and help structure findings which may have relevance to his design.

The study by [29] shows ill-effects of password policies which do not consider the users capabilities (like the ability to memorize passwords) and thus motivates the design of an artifact to consider or overcome these disadvantages. As such the requirements are constructed (or adjusted) in order to eliminate the cause (bad password minimum specifications), so the undesired effect (user coping mechanisms which endanger security) does not set in or is at least mitigated.

The knowledge that multiple passwords are a burden for users can lead to adjustments of the policy in regard to whether or not the user should be forced to have unique passwords for every system he has access to. Coupled with the knowledge that frequent password changes have a negative effect this can lead to the design decision, that users should only use one password (high memorability) with a moderate to low change frequency – which is balanced by stricter rules regarding the password strength. When formulating a policy the implied goal is that it will be followed, so the desired effect (utility of the artifact "password policy") sets in.

Table 2. Impact of Episteme on the Design of a Password Policy

Source of Episteme	**Empirical Findings**	**Application of Empirical Knowledge**	**Impact on Techné**	**Design Process Step(s) [19]**
[29] (Reference theory from artificial phenomena)	Conflicts between policies and user capabilities Coping mechanisms of users endangers overall security	Motivating artifact construction (#1) Guiding design decision and action (#2)	Requirement: "Password rules must match user capabilities" Configure policy rule (component) to allow for long password lifetime before cycling.	Problem identification and motivation Define the objectives for a solution Design and development
[30] (Reference theory from artificial phenomena)	High level of perceived sanction severity increases user compliance with security policies	Guiding design decision and action (#2) Evaluate design result (#3)	Requirement: "Policy compliance" Configure the policy-component "Sanctions" to reflect high severity Utility: Consider costs of high sanction severity (e.g. recruitment costs if sanction is job termination)	Design and development Evaluation
[31] (Reference theory from natural phenomena)	Moral beliefs are a strong determinant of deviant behavior and work as a self-regulatory approach	Guiding design decision and action (#2)	Phrase the guidelines (component) in a way which appeals to moral beliefs	Design and development
[32] (Reference theory from artificial phenomena)	Educating users on password selection methods improves password quality Random passwords are harder to remember than other types	Guiding design decision and action (#2)	Requirement: "Policy should also educate users" Add mnemonic passwords and pass phrases as recommendations in the component "Guidelines"	Define objectives for a solution Design and development

With the knowledge from the work of [30] such compliance can be improved by configuring the sanction component of the policy with a high (perceived) sanction severity. Whether or not side-effects occur and if they outweigh the benefits needs to be evaluated. Additionally the study by [31] can lead to the insight and design decision that a linguistic framing around morality can serve as a self-regulatory approach for users who consider deviation from their moral beliefs to be taboos. [32] show the effects of various password selection methods to password strength and password memorability. It can help improve a design by striking a balance between computational password strength and users ability to memorize, which – as we learned from [29] – decreases the need for subvert user coping mechanisms.

5 Discussion and Limitations

Many arguments in the ongoing philosophical exchanges about behavioral and design science are tied to the rigor-versus-relevance debate. A certain polarization of the debate has led to a wide-spread belief that behavioral science research is rigorous, but lacks (practical) relevance and merely triggers “so what?” reactions. Design science research on the other hand has been strongly characterized as being relevant, which is allegedly attained through a disregard for scientific rigor. We believe that these allegations are true to a certain degree.

However, the labels and strong polarization (and thus the cliff created between these two science paradigms in ISR) is hindering our ability to improve either – whether to make behavioral science more relevant or making design science more rigorous. In our research we took a new – to some extend naïve – look at the two paradigms based on the type of knowledge both generated from and required for research in the respective fields.

Our framework structures sources of different types of knowledge, their possible interaction with each other and allows in conjunction with e.g. the design science process model by [19] a structured enrichment of the design science research process with empirical knowledge. We believe such a design process to be more rigorous from a design science perspective. At the same time it allows behavioral scientists to showcase the relevance of their work by proxy of design artifacts.

Four points of where empirical knowledge can be injected into design science research have been identified. Using well-established work in this field we anchored these types of application with references in [2] and [19]. We imagine two possible types of use for our framework: 1) planning and structuring of design science research and 2) evaluation of design science research (with a focus on the rigor). The brief example gave a glimpse in how the framework can help with planning and structuring the design of an artifact. Our framework does not give guidance on what steps constitute design science research. Instead we show what types of empirical knowledge can be used in what phase of designing an artifact.

Another way of applying the framework is to evaluate design science research with it. In this scenario the four types of application of empirical knowledge can be used checklist-like. E.g., it can be used to determine if there is enough empirical evidence for the problem statement, which drives the motivation for the design of an artifact.

There are, however, limitations to the presented research, which we need to point out. We believe the philosophical discussion regarding episteme and techné to be sound. However, we did not introduce the three other Aristotlean virtues of thought phronesis, sophia, and nous. Therefore, we do not e.g. refer to ethical aspects or experience (in the sense of wisdom) although these cannot be deemed irrelevant, especially not in design science research practice.

Additionally, while we describe integration points with work by [2] and [19], we have not yet tested and evaluated the extent of possible integration between our framework and the design science process model or the design science guidelines. While we appreciate the ability to stand on the proverbial broad shoulders by "sourcing" both the process and normative component, it remains to be analyzed whether these two components need to be adapted for the specifics of rigorously empirical knowledge driven design science research.

Lastly, our presented example is brief in nature and a more comprehensive case study would be desirable to demonstrate the application of the framework for the structuring of design science research. Also an additional example showcasing the ability to evaluate design science research is required.

6 Concluding Remarks and Future Research

The goal of our paper was to examine how the interplay between behavioral (empirical) and design-oriented research (including artificial phenomena) can be improved from a design science perspective in ISR. Our research led to the distinction between empirical and theoretical knowledge (episteme) as well as non-artifact-centric and artifact-centric knowledge. To the best of our knowledge, a slightly related view can only be found in [20], what she calls theorizing about artifacts in use, and vaguely also in [6], what they call design knowledge.

Our framework allows the structuring of design science research and supports the design practice with empirical guidance. We claim that the application of theoretical and empirical knowledge in design science can improve the rigor of research. Our framework gives an idea of how behavioral IS knowledge, which frequently is criticized as lacking relevance [1, 33], can gain significant relevance by proxy; through its application in design science. This is done by aiding the problem definition and motivating the design of an IT artifact as well as supporting design decisions. There are still some open questions left which imply the need for further research. The explicit view of artifact-centric research as an own research stream is quite new. Therefore, it is unclear whether the outputs of this research are theories about artifacts or just a kind of weaker knowledge comparable to technological rules. Additionally, the question might arise who has to perform this research. Does it belong to the domain of design science or is it research in the behavioral science? We also recognize a need for more research on the four types of application of empirical knowledge, because no claims on the completeness of our list can be made as of now. Additionally, a deeper analysis of the characteristics of the different knowledge types might enhance our understanding of the interplay between the four quadrants.

References

1. Frank, U., Schauer, C., Wiegand, R.T.: Different Paths of Development of Two Information Systems Communities: A Comparative Study Based on Peer Interviews. Communication of the AIS 22, article 21, 391–412 (2008)
2. Hevner, A.R., March, S.T., Park, R., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28(1), 75–105 (2004)
3. Österle, H., Becker, J., Frank, U., Hess, T., Karagiannis, D., Krcmar, H., Loos, P., Mertens, P., Oberweis, A., Sinz, E.J.: Memorandum on Design-Oriented Information Systems Research. European Journal of Information Systems 20(1), 7–10 (2011)
4. Winter, R.: Design Science Research in Europe. European Journal of Information Systems 17(5), 470–475 (2008)
5. Van Aken, J.E.: Management Research Based on the Paradigm of the Design Sciences: The Quest for the Field-Tested and Grounded Technology Rules. Journal of Management Studies 41(3), 219–246 (2004)
6. Fettke, P., Houy, C., Loos, P.: On the Relevance of Design Knowledge for Design-Oriented Business and Information Systems Engineering. Business & Information Systems Engineering 2(6), 347–358 (2010)
7. Goldkuhl, G.: Design Theories in Information Systems – A Need for Multi-Grounding. Journal of Information Technology Theory and Application 6(2), 59–72 (2004)
8. Nunamaker, J.F., Chen, M., Purdin, T.D.M.: Systems development in Information Systems Research. Journal of Management Information Systems 7(3), 89–106 (1990)
9. Walls, J., Widmeyer, G., El Sawy, O.: Building an Information System Design Theory for Vigilant EIS. Information Systems Research 1(3), 36–59 (1992)
10. March, S., Smith, G.: Design and Natural Science Research on Information Technology. Decision Support Systems 15(4), 251–266 (1995)
11. Hirschheim, R.A.: Information Systems Epistemology: An Historical Perspective. In: Mumford, R., Hirschheim, R.A., Fitzgerald, G., Wood-Harper, A.T. (eds.) Research Methods in Information Systems, pp. 13–35. Elsevier Science Publishers, Amsterdam (1985)
12. Aristotle: Nichomachean Ethics. New York, Macmillan (1990)
13. Parry, R.: Episteme and Techne. The Stanford Encyclopedia of Philosophy, `http://plato.stanford.edu/archives/fall2008/entries/episteme-techne/` (accessed on February 7, 2011)
14. Agazzi, E.: From Technique to Technology: The Role of Modern Science. Society for Philosophy and Technology 4(2) (1998)
15. Baskerville, R., Pries-Heje, J.: Explanatory Design Theory. Business & Information Systems Engineering 2(5), 271–282 (2010)
16. Goodmann, N.: Fact, Fiction, & Forecast. Harvard University Press, Cambridge (1955)
17. Fischer, C., Winter, R., Wortmann, F.: Design Theory. Business & Information Systems Engineering 2(6), 340–387 (2010)
18. Simon, H.: The Sciences of the Artificial, 3rd edn. MIT Press, Cambridge (1996)
19. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems 24(3), 45–77 (2008)
20. Gregor, S.: Building Theories in the Science of the Artificial. In: 4th International Conference on Design Science Research in Information Systems and Technology, Pennsylvania, Philadelphia (2009)
21. Hooker, J.N.: Is Design Theory Possible? Journal of Information Technology Theory and Application 6(2), 73–83 (2004)

22. Kuechler, B., Vaishnavi, V.: Theory Development in Design Science Research: Anatomy of a Research Project. European Journal of Information Systems 17(5), 489–504 (2008)
23. Gregor, S.: The Nature of Theory in Information Systems. MIS Quarterly 30(3), 611–642 (2006)
24. Goeken, M., Patas, J.: Evidence-based Structuring and Evaluation of Empirical Research in Requirements Engineering. Business & Information Systems Engineering 2(3), 175–185 (2010)
25. Goldkuhl, G., Lind, M.: A Multi-Grounded Design Research Process. In: Winter, R., Zhao, J.L., Aier, S. (eds.) DESRIST 2010. LNCS, vol. 6105, pp. 45–60. Springer, Heidelberg (2010)
26. Gregor, S., Jones, D.: The Anatomy of Design Theory. Journal of the AIS 8(2), 312–335 (2007)
27. Venable, J.: The Role of Theory and Theorising in Design Science Research. In: 1st International Conference on Design Science Research in Information Systems and Technology, Claremont, California (2006)
28. Hevner, A.R., Chaterjee, S.: Design Research in Information Systems – Theory and Practice. Springer, New York (2010)
29. Inglesant, P.G., Sasse, M.A.: The True Cost of Unusable Password Policies: Password Use in the Wild. In: 28th International Conference on Human Factors in Computing Systems, Atlanta, Georgia, pp. 383–392. ACM, New York (2010)
30. Herath, T., Rao, H.R.: Protection Motivation and Deterrence: A Framework for Security Policy Compliance in Organizations. European Journal of Information Systems 18(2), 106–125 (2009)
31. Wenzel, M.: The social side of sanctions: Personal and Social Norms as Moderators of Deterrence. Law and Human Behavior 28(5), 547–567 (2004)
32. Yan, J., Blackwell, A., Anderson, R., Grant, A.: Password Memorability and Security: Empirical Results. IEEE Security & Privacy 2(5), 25–31 (2004)
33. Benbasat, I., Zmud, R.W.: Empirical Research in Information Systems: The Practice of Relevance. MIS Quarterly 23(1), 3–16 (1999)

Design Theory in Practice – Making Design Science Research More Transparent

Kalle A. Piirainen[1,2] and Robert O. Briggs[3]

[1] Advansis Ltd., Mechelininkatu 1a, FI-00180 Helsinki, Finland
kalle.piirainen@advasis.fi
[2] Lappeenranta University of Technology, Skinnarilankatu 34, FI-53850 Lappeenranta, Finland
[3] Center for Collaboration Science, University of Nebraska at Omaha, College of Business Administration, 6708 Pine Street, Omaha NE 68182-0048, The United States of America
rbriggs@mail.unomaha.edu

Abstract. Design Science Research seeks to develop new generalizable knowledge about design processes, design products, and designed artifacts while solving organizational problems with new work practices based on information technology. However, the ability of Design Science Research to generate knowledge has been challenged by some scholars, due weak connection of the designed artifact to the knowledge base. Design Theories offer a promising approach to codify and generalize some aspects of the knowledge created, in particular that pertaining to design processes and products for a given class of information technology-based solutions. We present a case example to support our argument that Design Theory can be integrated into the context of Design Science Research to make the connection between the design and the knowledge base more transparent, rendering it easier to defend the rigor and generalizablity of the knowledge Design Science Research yields.

Keywords: Design science; design science research; methodology; design theory.

1 Introduction

Design Science Research (DSR) [10], [18] in Information Systems (IS) research seeks to develop new generalizable knowledge about design processes, design products, and designed artifacts while solving organizational problems with new work practices based on Information Technology (IT). Although DSR is still relatively young as a research approach, practitioners have produced useful insights about, for example, managing documents [33], commercial samples [21], enhancing the business model for travel agencies [22], and designing collaboration systems [16].

Because DSR focuses on building artifacts that solve problems, it faces the important challenge of establishing and defending the rigor and generalizablity of the knowledge it yields. There are certain threats to the applicability of knowledge produced by DSR, for example, opaque grounding of the designs and a weak connection between the artifact and claimed contribution [13]. These challenges reduce

H. Jain, A.P. Sinha, and P. Vitharana (Eds.): DESRIST 2011, LNCS 6629, pp. 47–61, 2011.

the rigor of DSR and inhibit generalizing the results of apparently successful design projects back to the knowledge base.

One approach proposed to increase the rigor and generalizability of DSR findings is to codify certain aspects of new DSR knowledge as Design Theories (DTs) [8], [31]. A DT is a prescription for producing high-quality designs of a particular class of objects, defined as *"… a prescriptive theory based on theoretical underpinnings which says how a design process can be carried out in a way which is both effective and feasible."* [31, p. 37]. Gregor and Jones [8] recently proposed an anatomy for a DT that considers *design as an artifac*t. They [8] identify eight components of a design theory: (1) purpose and scope, (2) constructs, (3) principles of form and function, (4) artifact mutability, (5) testable propositions, (6) justificatory knowledge (kernel theories), (7) principles of implementation, and (8) an expository instantiation. Under this framing, a design theory is a collection of knowledge about how to discover, structure, and represent the knowledge embodied in the design.

We propose that the formalization of design knowledge in a DT codifies the properties and the foundations of the artifact to an easily transferrable form, renders evaluation more transparent and consequently helps to communicate the properties of the design. We argue that a DT can therefore be used by DS researchers as a framework to structure artifact descriptions so as to establish the scientific grounding of the artifact and to transparently link the artifact to the knowledge created by DSR. As evidence for this position, we present a study that exemplifies the development of explicit DTs during DSR. We show the utility of DT for improving rigor while increasing the transparency of DSR processes for designers and for readers of the DSR literature.

2 Recapitulating an Anatomy of a Design Theory

Most theories in the social sciences are meant to define relationships among constructs. The goal of such models is to explain observed variations in phenomena that manifest in a social setting in a way that would let one predict what would happen under a given set of conditions. A DT, on the other hand, prescribes a set of concepts and activities for designing some class of artifacts. It follows that DTs are theories of procedural rationality, to use Simon's [28] term. They provide ways to describe the artifact and its properties, and they prescribe a process by which the artifact can be designed and built. The concepts and activities that comprise a DT are often informed by one or more (grounded or nomological) 'kernel' theory [31].

Gregor and Jones [8] revise the Walls et al. [31] rendition of DT and simplify the structure, proposing that the design process and design product need not be separated, and that the same kernel theories often apply both to the process and the artifact. This is because a) the design is meant to improve some outcome of interest; b) the kernel theory predicts and/or explains the outcome of interest; c) the design is therefore a means to invoke more useful values of antecedent constructs in order to obtain better values of the consequent construct – the outcome of interest that the designer seeks to improve. The process specified in a DT, therefore, must suggest ways one can invoke better values of the antecedent construct. Therefore the kernel theory applies to both the process and the artifact. This change in perspective also moves the focus from prescribing design action to outcomes, which answers to Hooker's [12] critique, who posited that it is not possible to codify *design as action* by a covering normative design theory.

Gregor and Jones [8] also introduce three new facets to DT based on Dubin [5] (detailed in Figure 1 below). Firstly, the units or constructs that the theory deals with; secondly, artifact mutability, i.e. to what extent the theory predicts changes in the artifact when implemented or how the artifact could or should be changed from the initial rendition or instantiation; and thirdly, expository instantiation, a real-life proof-of-concept as an auxiliary component. Even though constructs are not explicitly addressed by Walls et al. [31], [32], we can assume that the meta-requirements or -design have to deal with constructs at least implicitly, so we can connect them. In a similar fashion, we can map mutability to meta-design as a researcher will, again at least implicitly, consider points for further research or how the artifact will fare after implementation. The comparison between the definitions (Figure 1) leaves one item, the proof-of-concept instantiation that is perhaps a nod toward Hevner et al. [11], who propose that DSR should produce an instantiation of the artifact.

3 Design Theory in the Design Science Methodology

While the concept of DT is clear, there is a need to relate the artifact to the DT to reconcile the DSR framework [9], [11] and DT. Walls et al. [31] propose that a design (theory) can be ultimately validated only through an artifact that can be observed and measured. The DT, they suggest, should prescribe what properties the artifact should have and how it should be built to solve the research problem [31], [32]. In a similar vein Gregor and Jones [8, p. 327] propose that "[t]esting theoretical design propositions is demonstrated through an instantiation by constructing … [an artifact]…" We can interpret that an artifact is an instantiation of a DT, which can be instantiated in the environment, e.g. a business organization.

We can also interpret that the artifact should embody or operationalize the DT to the extent that we can validate the theory by comparing the artifact informed by the theory to other artifacts or to work performed without the artifacts, thus achieving the research setting of comparing a treatment to other treatments and to non-treatment. This discussion also brings us to epistemology. As conceded in pragmatist philosophy, e.g. [14], which is compatible with, if not prevalent mode of thinking, DSR [29], a proposition is true if it works and is useful, so in extension, a design theory is valid, if acting upon it will produce an artifact that embodies or represents the justification knowledge and is useful in solving the original problem. So we should arrive at justified true beliefs if we follow Gregor and Jones' [8] prescription of constructing a DT and if we can show that it solves the research problem in a useful way. Thus we can draw from the structure of DTs to set the tasks for evaluation. The first purpose of evaluation is to examine the validity of the DT by verifying that following the principles of form and function set by the DT will produce an artifact that represents or embodies the justification knowledge or kernel theories, and secondly to evaluate whether it proves to be useful, just as discussed by James [14].

On a practical plane, Peffers et al. [23] have modeled a methodological guideline for organizing DSR complementing the 3-cycle model [9]. The initial phase of the DSR process is outlining the problem, which results in a research proposal. The second phase then concentrates on suggesting solutions to the problem defined in the proposal, where the knowledge base is accessed to find feasible solutions. The third phase is effectively the design phase. Here the researchers use the suggested solutions to develop or construct the artifact. The design is demonstrated and rephrased if necessary. After design and/or demonstration, the artifact moves into evaluation, where the "utility quality and efficacy of the artifact has to be rigorously evaluated" [11].

The methodology mirrors the proposed structure of DT. The definition of the research mission will firstly draw the outline for design context and meta-requirements. The objective definition can be a backdrop to define the constructs and scope of the theory. The design is based on the justification knowledge and addresses the principles of form, implementation and testable propositions. The demonstration can act as an expository instantiation and the evaluation will finally validate the DT.

Just as we can map the anatomy of a DT to the DSR methodology, we can also map its properties to the two challenges for rigor, which are opaque grounding of designs and weak connection of the artifact and justification knowledge [8], [13], as illustrated in Figure 1. The purpose and scope as well as the constructs guide the search within the knowledge base to find principles of form and function and to anticipate mutability of the artifact. In plain language, using the DT gives form to the rigor cycle [9] and the design product, the artifact, by grounding the reasoning behind the design and the general principles of form and function to literature. Further the DT can also support the design cycle as the DT is demonstrated, evaluated and rephrased based on the findings from the evaluation [23], [29]. The testable propositions have a dual function as they serve to ascertain that the instantiated artifact embodies the justification knowledge and that the design works as proposed. Thus the propositions ground the constructs and justification knowledge (theories), to the artifact, and empirical world, enabling claims to new knowledge. The DT itself ensures that the reasoning behind the design is transparently displayed and can be evaluated while considering the contribution of DSR. In sum, we may claim that the DSR methodology and DT complement the DSR framework and give additional guidance how to work within the borders set by the 3-cycle model.

Thus, the DT acts as an interface between the world and the knowledge base. The DT first codifies the base (kernel) theory and the principles of form and function, the essential shape of the artifact, thus connecting the design transparently to the knowledge base. Second, the testable propositions that are directly derived from the foundation of the design will serve as a basis for hypotheses to test whether the artifact instantiates the DT and whether it fulfills the requirements set for it. These propositions link the claims to knowledge to the existing knowledge together with the explicit constructs, scope and justification knowledge. On these premises we argue that the DT will improve transparency and rigor of DSR research.

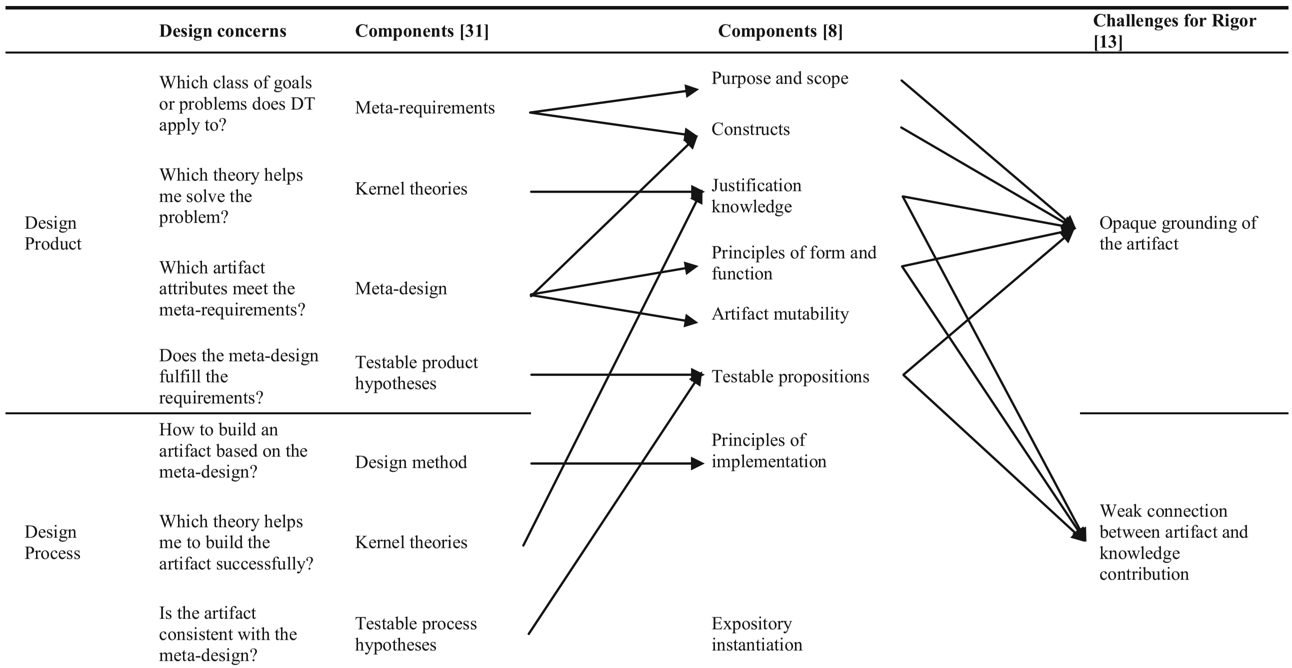


Fig. 1. Connections between the anatomy of a DT to rigor and transparency (adapted from [24], orig. [8], [30], [31], [32])

4 Case Illustration: Design of a Collaboration Process for Scenario Planning

This section illustrates the previous conceptual discussion through presenting a case example where the researchers chose to use a DT to enhance the research process. The description is a concise overview that is concentrated on the aspects related to the DT and is intended to give an overview of the research and the position the DT has in the design. While we will recapitulate and interpret the relevant details below, a full report with the minutes of the design and process are presented in [24].

4.1 Overview of the Design Process and Design Theory

The objective of the project was to design a scenario-based method to support strategic technology management. The field of technology management is risk laden and technology foresight can help to reduce technology related and business risk, e.g. [27]. Existing methods for scenario development were inefficient, resource intensive and not accessible to practitioners (see [25] for discussion). The goals were to increase the performance of scenario planning teams, as well as to to increase the transparency and explicitness of design and evaluation methods, making them more accessible, and making it possible to conduct a critical appraisal of the results. The research questions for the project were:

R1: What are the business need and challenges for scenario planning in strategic technology management?
R2: How we can improve the effectiveness of scenario planning process?
R3: How can these methods be implemented to improve the state-of-the-art of strategic technology management?

The meta-requirements for the DT were that the designed scenario method should be more accessible and effective while it should fulfill certain criteria for a successful scenario process. The design to solve the problem was an electronically mediated intuitive-logical scenario method, as summarized in Figure 2. The figure is a representation of the DT, which has several main components, including the scenario process, the electronically mediated expert panel workshop, and the intuitive impact-based scenario heuristic.

The actual DT is a composite of existing theories for explaining (and predicting) [7], prescriptive knowledge in the form of local and published best practices codified in Table 1 (below) that prescribes what are the basic principles of form and function that can be used to solve the design problem. The 'formal' DT summarizes the wider description how to build the artifact, but in the case, the DT in the widest interpretation spans tens of pages from the description for the problem to the description of the process and conclusions about the evaluation [24].

The research process followed the DSR methodology by Peffers et al. [23]. The limitations of existing scenario methods were already well scoped in the literature, so the problem was already well defined and the entry to the DSR process was solution oriented. The project therefore began with solution generation and design. The designers

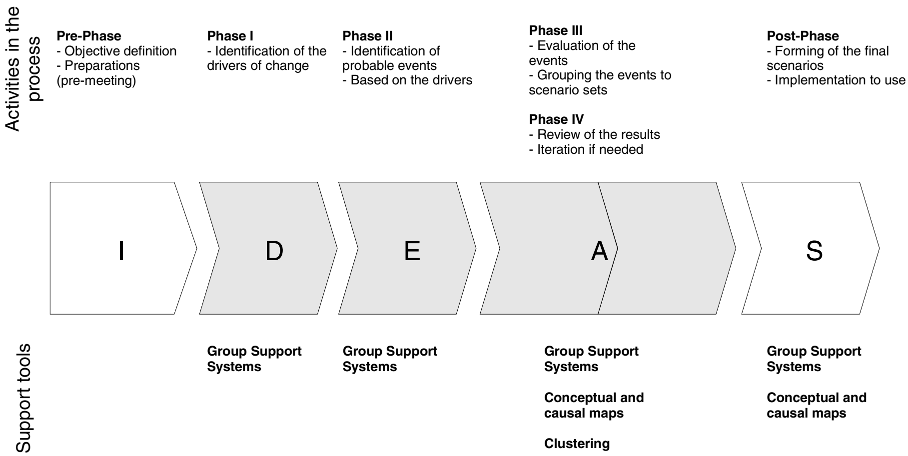


Fig. 2. Overview of the form of the design (source: [24])

had a viable technology to help solve the problem; a general purpose Group Support System (GSS), e.g. [26], with which they were already familiar. This system provided capabilities around which a sound collaboration process could be developed.

The knowledge base provided exploratory and applied science findings about how the technology had been used to support other work practices, and about how it could be used to create and sustain useful patterns of interaction among team members (e.g. to generate, reduce, organize, and evaluate ideas). The knowledge base also provided designers with field observations and best practices from the scenario planning literature, from which principles of form and function could be derived to guide the work practice design.

The basic principles of form and function were drafted based on the literature and the DT was instantiated in a solution. The design process was iterative, with several cycles involving a) defining a deliverable for the group; designing a sequence of activities for creating the activities; c) defining a sequence of collaboration patterns for each activity; d) scripting a procedure for each activity; d) configuring appropriate technical capabilities for each activity, e) pilot testing and evaluating the solution. After initial trials, key elements of the solution, most importantly the scenario heuristic, were modified and the solution was tested again.

Evaluation followed the Hevner et al. [11] framework, and McGrath's [20] notions about choosing research methodologies that balance representativeness, depth of understanding and generalizability. The main phases of the empirical evaluation of the artifact/DT were: 1) experimental evaluation of the meta-design with student groups, which resulted in revising the DT, 2) case-based evaluation of the artifact in two scenario processes, and 3) benchmarking with an established state-of-the-art, scenario method and with a similar electronically mediated process.

The first part of the evaluation focused on the early phases of the scenario process where the substance for the scenarios is gathered, as Piirainen [24] argues that the scenarios are only as good as the data that inform them, which makes the front-end of the process an important bottle neck. The rest of the evaluation relied on the pragmatist logic "what works is true" in the sense that as far as the artifact works and creates plausible scenarios comparable to competing artifacts, it satisfies the conditions for a pragmatic truth [14].

The first phase of evaluation tested the design in trial sessions with student groups executing an assigned task. These focused on the first design proposition (DP1) and

Table 1. Components of the final DT for the IDEAS method (adapted from [24])

	General components	Guiding question	Components of the DT
Core components	Purpose and scope	Which class of goals or problems does the DT apply to?	- Class of problems: organizational (technology) foresight with emphasis on resource use and efficiency of the foresight Goals: - The scenario process and method should be easily available, documented, and executable - Scenarios should be reliable, consistent and convincing
	Constructs	What are the key units and constructs governed by the theory?	- Events, drivers and scenarios - (The generic) Scenario process - Scenario heuristics - Support tools, GSS and mapping
	Justification knowledge	Which literature helps me to solve the problem by building an artifact?	- Theory-in-use from FS and scenario planning - IS research and GSS literature - Local and global best practices for facilitation
	Principles of form and function	Which (class of) artifacts meet the meta-requirements?	An artifact combining: - The generic scenario process, with - The intuitive impact-based scenario heuristics, and - Mapping tools and clustering
	Artifact mutability	How does the artifact behave when implemented?	- The contents of the workshop depend on the context - The content of the scenarios will affect the illustrations and tools used in the process - The scenario heuristics can be adapted to suit the case - The workshop does not have to be a facilitated face-to-face session - The method can be used on a variety of foresight problems

Table 1. (*continued*)

	General components	Guiding question	Components of the DT
	Testable design propositions	Which questions we need to answer to know that 1) the meta-design satisfices the requirements 2) operationalized the design knowledge, and 3) solved the problem efficiently?	DP1: The artifact is feasible and usable DP2: The artifact produces scenarios reliably DP3: The artifact enables effective production of scenarios DP4: The process is structured yet innovative DP5: The people are engaged and feel free to contribute DP6: The process compares favorably to existing scenario practices
Additional/ auxiliary components	Principles of implementation	How to build an artifact based on the meta-design?	Use the description of the meta-design with established best practices to build and execute the scenario process
	Expository instantiation	Is the artifact consistent with the meta-design?	Cases 1 and 2 instantiated artifact following the DT/meta-design

tested whether the basic design was a feasible solution to the problem. The tests followed an 'engineering method' [35], that is, multiple develop/test cycles until further improvement is not necessary. Study design for evaluation was *XO*, i.e. treatment and post test [3]. Trials were run in a laboratory. Process and products were evaluated with questionnaire. The questionnaire instrument measured satisfaction with process, tool usability, and scenario quality (see [24] for details). The feedback from the test subjects guided improvements to the DT, the final version of which appears Table 1 and a new artifact was derived from the improved DT.

The improved artifact was evaluated with two case-based tests. Case study research has a long tradition in IS as a strategy for exploratory research [1], [17], and for artifact evaluation [15], [2]. The protocol for the two cases was built on Yin [34] and Eisenhardt [6]. The first case was within a public organization, where the process ran its course all the way to finished scenarios. The first case was followed by a second case, which adapted the DT to a management of technology context. Both of the cases were aimed to test the repeatability and reliability of the DT and artifact (DP2) and to get a better view to the inner workings of the process (DP5; DP5) as well as to benchmark it against other, more established methods (DP6).

In the first case the test subjects were administrative and operational staff of the organization, and as such the case can be characterized as a production run more than as a simple laboratory experiment. In the second case the DT was instantiated in a workshop with industry experts and independent researchers in association with a research project, and the group developed technology scenarios. The case-based evaluation continued to a benchmarking study between the artifact and another scenario method. The benchmark studies were reflective comparisons between the DT and two scenario methods in two different contexts. The first benchmark was the field anomaly relaxation method [4] and the other was another GSS-driven scenario method called the SAGES method [25].

To summarize, the evaluation design was based on DSR framework and general IS and Management literature following ordinary good practices in the field. The DT was interfaced with evaluation through the design proposition which acted as sub-research question for evaluation as indicated above. During the research the DT (summarized in Table 1) was used to guide the design process together with the DSR methodology framework and to support the design iterations and evaluation. Overall the mission of the evaluation was to ascertain the utility, quality and efficacy of the DT to solve the design problem. The conclusion for this case research was that the design was successful as it worked as intended and the empirical testing provided evidence that the design propositions are fulfilled and the goals are achieved. In this sense the DSR process was complete, although the researchers explicitly stated that sustained adoption of the artifact was not a part of the enquiry and further that due to the limited field designs the research should be treated as an extended proof-of-concept rather than a full validation of the DT.

4.2 Reflections and Findings about the Design

As per the research mission of this paper, the description of the case illustrates two things about DSR. Firstly, following the Hevnerian [9] guidelines of DSR does not preclude using a DT as an additional tool for design or following a DSR 'methodology' as described by Peffers et al. [23]. And even more importantly, the case illustrates how a DT can support DSR throughout the process. Table 1 illustrates specifically how the DT framework summarizes the theoretical basis for the artifact transparently, and connects the evaluation of the DT or its instantiations to the design through the design propositions.

The great contribution of Hevner et al. [11] was to set up an open framework for DSR and to legitimize the mission in the field. Walls et al. [31] came up with the DT a dozen years prior to Hevner et al. [11], but the use of DT has been limited. While the DSR literature does not offer significant evidence that the field would divided to camps favoring either 'Hevnerian' 3-cycle model-based or 'Wallsian' DT-driven DSR, Walls et al. [32] note that after the first ten years since the publication of this seminal piece there was still quite few papers that actually used a DT as they proposed, although a large number of papers cite their article. As an explanation they [32], supported by others [10], present a finding that some scholars find the DT as cumbersome to use, and for example students have had troubles in grasping the basic concept of DT.

With the described case we answer to the call of widening the use of DTs [32] and demonstrate how to strengthen the DSR framework by adopting a DT as a guideline to compile the design and contribute back to the knowledge base. Our rendition of the DT framework is a synthesis of Walls et al. [31], Venable [30] as well as Gregor and Jones [8], with a quite strong emphasis on the latter source. We find that the DT acts as an axle to the design cycle, helping in codifying the design choices and making the theoretical basis and assumptions behind the design transparent. These properties make DT useful in contributing to the knowledge base. An artifact as a product of DRS is an instantiation of theories and, some argue, does not have truth value but only utility [19], while others like Iivari [13] concede that artifacts may be outside evaluation of truthfulness, but insofar as DSR is to add to the knowledge base, validation of theoretical insights that arise from the process is an issue. Although especially within pragmatist philosophy theoretical contribution or truthfulness is not often the main aim or research, IS field in general is committed to both advancing knowledge and solving business problems, and while an artifact can be a useful solution it is as such a weak addition to the knowledge base. A DT helps as it not only describes an instance of problem solving, it describes what class of problems can be solved with the DT, what are the theoretical basis and why the artifact is supposed to work, and how to build an artifact to solve that particular instance of the class of problems

What follows from formulating an explicit DT is, ideally, improved rigor and generalizability of DSR outputs. As Ann Majchrak (through [32]) proposes, a DT can provides a vehicle to "…articulate … contributions to readers and scholarly consumers with a common agreed-upon language which [is] recognizable and repeatable." In a word a DT gives a proper form for a contribution to the knowledge base, as it not only summarizes the conceptual foundations of an artifact in a wrapper that is reusable to a class of problems, it also helps to draw the borders of applicability for a DT as it explicitly illuminates the conceptual foundations and design assumptions. In fact, as discussed by Gregor [7] as well, while an artifact is not necessarily a contribution to the knowledge base however useful it might be, a design theory on the other hand is, as it clearly defines what is the phenomenon(-na) of interest, the constructs the theory governs and the inferences between the construct in a form that can be generalized to a defined population or class of problems.

Rigor in DSR is about ascertaining that we operationalize the principles of form and function sufficiently to embody the base theory in the artifact, so we can actually tell whether the theoretical design works. Here the structure of the DT lends a hand, as one specifies the theories the design is based on, what are the expected inferences and formulates refutable propositions that are tested empirically to evaluate the DT. The structure of a DT flows like a waterfall and it is relatively easy to keep track of the overall picture of the design and to check that all the pieces fit together. Additionally DT also removes opaqueness from grounding of the knowledge contribution to the artifact and the world, as the DT explicitly spells the justification knowledge, principles of form and testable design propositions that link the artifact to the claims to knowledge. While all this can be accomplished without a DT, the added benefit is that using an explicit 'template' such as a DT to achieve the documentation, makes the DSR process all the more transparent and repeatable.

Going deeper into the interaction of the conceptual elements, we can roughly map the components of a DT to the design process: the purpose and scope of the DT is determined by the design problem, the constructs are the building blocks to solve the problem one finds while searching the knowledge base, and justification knowledge for the design is an additional accumulation around the base constructs. The justification knowledge is used to formulate the principles of form and function during the design and artifact mutability is anticipated in design and demonstration. In fact the design cycle is an iterative loop between searching justification knowledge, improving the form and assessing mutability, until a satisficing DT is reached. During this design and demonstration, the testable design propositions can be formed following the key objectives of the design to guide the final evaluation of the DT. As we see, the process and DT interact as one builds the design, and a DT codifies the design choices, making them explicit.

With all this said, it has to be remembered that a DT is not a panacea to all design problems or to improving rigor of DSR. While a useful framework, a poorly constructed DT or a DT that does not reflect the reality of the design is little more useful than any other representation of a design. Looking at the research problem of the paper, opaque grounding of DSR and poor transparency of evaluation, there is a demand to craft the DT quite rigorously. To return to the reasons why the DT has been underutilized, our experience collate with the literature saying that the initial structure of DT was ambiguous and relatively complex [10]; there was a learning curve in using a DT. One factor was that the literature in general strongly favors the DSR framework, and the previous work that exists has done relatively little to make DT actionable within the existing framework. On a critical note, it pays to remember when constructing a DT that it adds little transparency if it fails to illuminate what is rationale and the conceptual basis of the design. Especially regarding the grounding of knowledge contributions the design propositions are the link that interfaces the claims to knowledge to the world. For example, the propositions in the presented case are rather general; to gain more detailed information about the strengths and weaknesses of the DT in solving the problem, one should be careful to devise propositions that are intimately linked to the design goals and specifically to the possible points of failure which may be revealed during the design-demonstration cycle. Further, when aiming for a solid theoretical contribution, the propositions have to be also clearly linked to the (kernel) theories and justification knowledge, to enable linking the empirical finding to the conceptual basis.

While discussing rigor, we need to recognize the limitations of the case example: starting from the experimental setting used in the tests, to the sample sizes and the extreme limitations they set for the statistical analysis of the data from a classical positivist perspective. In the example the researchers call the experiments 'tests' and the research an extended proof of concept, while they conclude that the DT and the instantiated artifact appear to be useful and to follow the justification knowledge. Some of the limitations can be regarded as oversights from the researchers part or as a sign of yielding to research economies, but also to the pragmatic(-ist) or instrumentalist orientation embedded in the research design. A crucial point is that the example was not chosen for its exemplary design and field methods, but because it illustrates the use of a DT to structure the design cycle within the DSR framework. So, in fact, the limitations of the example should not detract attention from the idea of using DT for structuring the design together with the DSR methodology within the DSR framework. The same logic and structure can be applied with a more rigorous evaluation design and more intricately derived design propositions to make an even greater contribution to the knowledge base.

To summarize, while we recognize the research design in the example is limited, it does not make the case less of an example of a relatively successful DSR project where the use of a DT added value as a device for codifying the design and proving a template for evaluating it. In fact, we might be inclined to argue that a DT might be used to operationalize for example a social science theory, a theory for explaining (and predicting) by constructing a DT based on it, to test it in the environment through instantiations. In this setting justification knowledge is what is tested and the surrounding DT documents the operationalization of that theory explicitly, giving clear borders of applicability and generalizability.

What these findings and reflection amount to, is the general claim that formulating a DT will be a valuable addition to DSR also within the general DSR framework. As already proposed in the introduction in this paper, we claim that the use of a DT to complement the DSR framework adds to the transparency and rigor of research. However, to be fair, we must remember that a DT is not an instant remedy and has to be rigorously constructed and used to add any value to existing practices.

5 Conclusions

The mission of this paper was to show through an example how a DT can be leveraged to improve the transparency and rigor of DSR. Our paper is positioned within the DSR framework presented by Hevner et al. [11], complemented with the DSR methodology by Peffers et al. [23] and our rendition of DT, based heavily on Gregor and Jones [8]. We started by discussing the DT within the foundations of DSR and continued on to present an example of a research that combines the conceptual elements successfully, creating added value.

This paper contributes to the IS literature by describing how codifying the foundations and properties of a design to a DT can be used to make DSR more transparent and rigorous. It follows that the DT can be leveraged to alleviate the opaque grounding of designs to the knowledge base by explicitly crafting a DT, and to improve rigor by using the DT further to link the previous knowledge explicitly to the artifact. In fact we may go as far as proposing that a DT can be used in research as a template for operationalizing theories to be tested in e.g. (quasi-) experiments.

As illustrated by the conceptual discussion and the case example, DT interfaces both with the DSR framework and the DSR process. Specifically the DT interfaces with the design cycle as it provides a framework to organize the access to the knowledge base for justification knowledge, to designing the principles of form and function to solve the design problem and to guiding the evaluation. The role of the DSR process or methodology [23] is to act as a practical guideline together with the general framework and the DT.

In sum, we find from the example that DT is a valuable addition to the DSR framework. The main lesson for practice is that the DT was, in our experience, useful in structuring the both the design process and product, and the completed DT helps to improve the transparency of DSR research and connects the claims to knowledge better to the world and the knowledge base. The key is not so much following the DT to the letter, but using it as a guide to structure the research process and a reminder to consider the relevant aspects of design. We argue that the case example goes a long way to show how a DT can be used to enhance the rigor and transparency of DSR, and how a DT fits in the DSR framework and how it is compatible also with practical research following a DSR methodology. The main advantage is that the DT can be used to structure the design cycle and to document the design transparently. Additionally the DT makes the operationalization of the research more transparent and helps to assess the applicability of DSR. Together these findings let us conclude that DT can help DSR to fulfill the dual mission of IS by making the design process more transparent and by helping the communication of the results practitioners and scholarly audiences alike.

References

1. Benbasat, I., Goldstein, D.K., Mead, M.: The Case Research Strategy in Studies of Information Systems. MIS Quarterly 11(3), 369–386 (1987)
2. Bragge, J., Merisalo-Rantanen, H.: Engineering E-Collaboration Processes to Obtain Innovative End-User Feedback on Advanced Web-Based Information Systems. JAIS 19, 196–220 (2008)
3. Campbell, D.T., Stanley, J.C.: Experimental and Quasi-experimental Designs for Research. Houhgton Mifflin, Boston (1966)
4. Coyle, G.: Field Anomaly Relaxation (FAR). In: Glenn, J.C., Gordon, T.J. (eds.) The Millennium Project Futures Research Methodology Version 3.0, AC/UNU Millennium Project (2009)
5. Dubin, R.: Theory development. The Free Press, New York (1978)
6. Eisenhardt, K.M.: Building Theories from Case Study Research. The Academy of Management Review 14(4), 532–550 (1989)
7. Gregor, S.: The Nature of Theory in Information Systems. MIS Quarterly 30(3), 611–642 (2006)
8. Gregor, S., Jones, D.: The Anatomy of Design Theory. JAIS 8(5), 312–335 (2007)
9. Hevner, A.R.: A Three Cycle View of Design Science Research. SJIS 19(2), 39–64 (2007)
10. Hevner, A.R., Chatterjee, S.: Design Research in Information Systems. Integrated Series in Information Systems, vol. 22. Springer, Heidelberg (2010)
11. Hevner, A.R., Ram, S., March, S.T., Park, J.: Design Science in Information Systems Research. MIS Quarterly 28(1), 75–105 (2004)
12. Hooker, J.N.: Is Design Theory Possible? JITTA 6(2), 73–83 (2004)
13. Iivari, J.: A Paradigmatic Analysis of Information Systems as a Design Science. SJIS 19(2), 39–64 (2007)
14. James, W.: Pragmatism, Dover Thrift Editions. Dover Publications, New York (1995)
15. Kitchenham, B., Pickard, L., Pfleeger, S.L.: Case Studies for Method and Tool Evaluation. IEEE Software, 52–62 (July 1995)
16. Kolfschoten, G.L., de Vreede, G.-J.: A Design Approach for Collaboration Processes: A Multimethod Design Science Study in Collaboration Engineering. JMIS 26(1), 225–256 (2009)
17. Lee, A.S.: A Scientific Methodology for MIS Case Studies. MIS Quarterly 13(1), 33–50 (1989)
18. March, S.T., Smith, G.: Design and Natural Science Research on Information Technology. DSS 15(4), 251–266 (1995)
19. March, S.T., Storey, V.C.: Design Science in the Information Systems Discipline: An Introduction to the Special Issue on Design Science Research. MIS Quarterly 32(4), 725–730 (2008)
20. McGrath, J.E.: Dilemmatics: The Study of Research Choices and Dilemmas. Am. Behavioral Scientist. 25(2), 179–210 (1981)

21. Moon, K.L., Ngai, E.W.T.: R&D Framework for and intelligent fabric dample management system: A design science approach. I. J. of Op. & Prod. Management 30, 721–743 (2010)
22. Novak, J., Schwabe, G.: Designing for reintermediation in the brick-and-mortar world: Towards the travel agency of the future. Electron Markets 19(15), 15–29 (2009)
23. Peffers, K., Tuunanen, T., Rothenbergerm, M.A., Chatterjee, S.: A Design Science Research Methodology for Information Systems Research. JMIS 24(3), 45–77 (2008)
24. Piirainen, K.A.: IDEAS for Strategic Technology Management: Design of an electronically mediated scenario method (Diss.), Acta Universitatis Lappeenrantaensis 406. Lappeenranta University of Technology, Lappeenranta (2010), http://urn.fi/URN:ISBN:978-952-214-998-5 (accessed 2.3.2011)
25. Piirainen, K., Lindqvist, A.: Enhancing business and technology foresight with electronically mediated scenario process. Foresight 12(2), 16–37 (2010)
26. Power, D.J.: Decision Support Systems, Concepts and Resources for Managers. Quorum Books, Westport (2002)
27. Schoemaker, P.J.H.: When and How to Use Scenario Planning: A Heuristic Approach with Illustration. J. of Forecasting. 10(6), 549–564 (1991)
28. Simon, H.A.: The Sciences of the Artificial, 3rd edn. The MIT Press, Cambridge (1996)
29. Vaishnavi, V., Kuechler, W.: Design Research in Information Systems. IS WorldNet, Atlanta (2004), http://ais.affiniscape.com/displaycommon.cfm?an=1&subarticlenbr=279 (accessed 2.3.2011)
30. Venable, J.R.: The Role of Theory and Theorising in Design Science Research. In: The Proceedings of Design Research in Information Systems and Technology Conference DESRIST 2006, Claremont (2006)
31. Walls, J.G., Widmeyer, G.R., El Sawy, O.A.: Building an Information Systems Design Theory for Vigilant EIS. ISR 3(1), 36–59 (1992)
32. Walls, J.G., Widmeyer, G.R., El Sawy, O.A.: Assessing Information System Design Theory in Perspective: How Useful Was Our Initial Rendition? JITTA 6(2), 43–58 (2004)
33. Wu, H., Gordon, M.D., Fan, W.: Collective Taxonomizing: A collaborative approach to organizing document repositories. DSS 50(1), 292–303 (2010)
34. Yin, R.K.: Case Study Research: Design and Methods, 2nd edn. Sage Publishing, Thousand Oaks (1994)
35. Zelkowitz, M.V., Wallace, D.R., Binkley, D.W.: Experimental validation of new software technology. In: Juristo, N., Moreno, A.M. (eds.) Lecture Notes on Empirical Software Engineering, pp. 229–263. World Scientific Publishing, River Edge (2003)

Harnessing Handheld Computing – Managing IS Support to the Digital Ranger with Defensive Design

Bo Andersson

Lund University, School of Economics and Management, Ole Römers väg 6, 23263 Lund, Sweden
bo.andersson@ics.lu.se

Abstract. The recent years of development in mobile computing as powerful handheld computers and high-speed wireless networks creates opportunities for new user-groups in the mobile workforce to take advantage of mobile technologies. User-groups may be more or less geographical distributed and as a consequence more or less marooned when it comes to obtaining IT/IS support and this increases the complexity of delivering IT/IS support to these geographically distributed end-users. In this design paper the aim is to develop a design theory to manage problems in IT/IS support to the outbound user. Semistructured interviews were performed with developers and documents studies of an information system comprising handheld mobile computing devices for drivers. From the interviews, a design theory based on the implemented strategy of defensive design is presented. The six components of IS design theory by Gregor and Jones is applied as a theoretical framework for evaluation of the design theory.

Keywords: Mobile Information Systems, Digital Rangers, Design Theory, Design Research, Remote Mobile Users, Support.

1 Introduction

The recent years of technical development in mobile computing creates opportunities for new user-groups in the mobile workforce to take advantage of mobile technologies. The benefits of these technologies, and thus developed information systems, are widely discussed [1, 2] in addition to the specific physical features of small handheld computing devices, such as the small form factor, wireless connectivity, security issues, see for instance Maunuksela and Nieminen [3], Satyanarayanan [4], Shin [5], Tarasewich et al. [6]. These mobile information systems allow for end-users that are more or less completely physically detached from their organisations and making them digital rangers, always ranging their territory between assignments. As a consequence of this detachment, the requirement for IT/IS support changes compared to IT/IS support for stationary users and places demand for either appropriate distant support or extra reliable mobile information systems. The basic assumption is that the task to deliver support is different if the user is stationary (i.e home based) or mobile (i.e outbound). That is, how to design and manage support for stationary users that operate for example within a building is different compared to support for remote

H. Jain, A.P. Sinha, and P. Vitharana (Eds.): DESRIST 2011, LNCS 6629, pp. 62–76, 2011.

users that are detached from their home base. The actual geographical distance makes support expensive if performed by personal interaction, and cumbersome due to the limited possibilities of interaction. In this paper the argument put forth is that information systems for the outbound user may be designed in order to minimise the need for support activities and to increased perceived quality of support. There have been a considerable number of works on end-user support for the stationary user see for instance Bergeron et al. [7], Doll and Torkzadeh [8], Mirani and King [9], Rockart and Flannery [10], Shaw et al. [11], Shaw et al. [12], Torkzadeh and Doll [13], however, how to manage efficient IS/IT support to the outbound mobile workforce, the digital ranger, is absent.

1.1 Objectives

Advancements in mobile technology, as more and more powerful handheld computers and the increased geographical coverage of high-speed wireless networks, are parts of the technological foundation for the development of a range of applications for the mobile workforce. In the current situation design research, informing developments on lessons learned from successful design, ought to have high relevance, as this knowledge may, when adapted to design propositions, enhance the systems to be built. Therefore, the scientific perspective applied is design science and an effect of this is the normative, prescriptive approach compared to a more traditional descriptive approach as in natural or social sciences [14]. A fundamental assumption in this paper is that the established design patterns and methods used when developing desktop (i.e. stationary) applications may be inappropriate when developing applications for mobile computing devices due to, for example, contextual concerns such as dynamic use situations and the "small form factor" [15, 16]. Another interesting remark supporting this assumption is made by Duchamp [17] in that mobile information systems are probably the first systems developed for others rather than ourselves. That is, the majority of previous systems are designed by knowledge workers for other knowledge workers. However nowadays systems for a wider user group are being built, blue collar users with no, or little, computing experience putting forward new demands on the design of the systems [17]. This renders a certain interest on expanding the design space to also embrace such as handheld computers and their recipients, the mobile workforce.

The lion's share of work done on new and innovative applications within the domain of mobile computing and mobile information systems has been with an technology-push stance, visible in for example Nichols and Meyers [18] or Toye et al [19], rather than a market-push [20]. Technology-push typically represents a situation where a (technical) solution exists and looks for problems to solve, the market-pull situation is the opposite where problems exist and (technical) solutions are wanted [21, 22]. In this paper the rhetoric is market-pull, in other words, an aspect on mobile computing (managing the support to outbound users) is problematic, let's investigate if some design consideration in the design and development phases can help to solve this problem.

Some clarification on the terms used is helpful to the reader of this paper: The mobile device in this context is a handheld computing device in the form of a mobile phone with computing ability (often labelled smart phone). Henceforth the abbreviation of the concept "handheld computing device" (HCD) will be used in the text illustrating a small form factor device with computing and wireless networking abilities and the term mobile information system depicts a computerised information system with parts that are mobile, i.e. HCD.

1.2 Case Settings

The findings are based on a case study of the development and use of an information system utilised by the County Council of Kalmar in Sweden. In Sweden the County Councils offer service travel to its citizens. A service travel is a transportation of people to and from health care units in the county, not to be mistaken with ambulance transport. Service travel only manages non-urgent travel, like scheduled medical consultations or an appointment with an optician, for those who cannot manage to get there on their own. Local transportation firms such as taxi firms or bus companies carry out the actual transportation. To manage the vehicles a logistic system, WinHast, was developed. The reason to develop WinHast was to reduce the cost of transportation by better utilisation of passenger capacity in cars and busses. There were considerable redundancies or overlap in transports. Cars often transported only one passenger at the time and this was considered a major problem to solve.

WinHast is an information system that monitors all vehicles on duty. Approximately 210 vehicles are equipped with HCD´s connected to the WinHast system. WinHast monitors the vehicle's position via GPS, logs the mileage, destinations and a set of important parameters such as connectivity, keystrokes, charging, et cetera, a billing function is also included, calculating mileage, time and fares. A central dispatch in the municipality of Högsby administrates the bookings and invoicing. One of their tasks is to optimise the transportation in order for the vehicles to transport as many clients as possible at the same time. This is made by route optimising and just-in-time adjustments on planned routes and the problem with low utilisation of transport capacity is considered, by the manager of the County Council of Kalmar, solved.

1.3 Outline

The paper is organised as follows: it begins with a short background where the circumstances of mobility and support are illustrated. Thereafter the research approach and findings from the case are introduced, followed by a presentation of the theoretical framework applied and the design proposition is put forward. Finally some conclusions and suggestions for future work complete the paper.

2 The Mobile Workforce

This article revolves around the user being a part of the mobile workforce. The concept "mobile workforce" may be an ambiguous concept and some clarification can be useful. Starting with "mobile" in "mobile workforce", herein it should be understood to be a user that is more or less outbound with little physical connection to the organisation to which the user belongs. The classes of mobility depicted by Dahlbom and Ljungberg [23] with local and remote mobility is expanded by the category of the "digital ranger" as a user class and "remote work site" as a type of place (see table 1). The reason for this extension is to illustrate the user class that are supposed to always be out on the field. It may be the ambulatory repairman receiving assignments via email or phone, leaving home to go directly to the customer and almost never going to the organisation's facilities. Or being a truck driver on the road for long periods of time, detached from any home base (the truck is not considered a base). The "remote work site" illustrates an ad hoc work site, the user cannot anticipate anything concerning support technologies or field use conditions. It may be a construction site where a client's forklift needs repair or along the roadside where some field repair is needed. This should be compared to the "remote base" in Dahlbom and Ljungbergs framework [23] concept where it illustrates another base with some anticipated support technologies and supporting colleagues. It may be another health care facility or a different office in another city. These "rangers" may use computerised mobile information systems and being more or less completely physically detached from their organisations making them "digital rangers", always ranging within their territory between assignments. Being a digital ranger renders differences compared to the other user-groups as most likely a lack of supporting colleagues and the absence of naturally occurring conversation and knowledge transfer [24] also the lack of supporting technologies such as copiers, servers, printers, et cetera, is a part of the ranger's work circumstances [25, 26]. These differences reduce the potential to communicate with, and advice the digital ranger concerning matters related to IS support. Advising, that can have the nature of knowledge transfers between colleagues and support via additional devices.

Table 1. The framework of mobility modes [23] extended with two more classes marked with *, these two encompass the digital ranger, the ever outbound user and the digital ranger's relation to the work sites

Place → User↓	Home based	Around home base	During transport	At remote base	At remote work site*
Stationary	X				
Wanderer	X	X			
Traveller	X	X	X		
Visitor	X	X	X	X	
Digital Ranger*			X		X

Continuing with the latter part of the concept "mobile workforce", "workforce" indicates that we most likely are dealing with a mandatory use and strong dependency on application. This is, depending on the type of information system the user may have, no options other than to use the offered information system exists. Of course, when dealing with email or similar applications there probably are alternatives, however if the information system is tailor-made there are no alternatives. If the computerised mobile information system completely replaces the previous information system, the user relies heavily on a well functioning system. If the system malfunctions, the user may run into more serious problems affecting efficiency and work output, compared to the stationary user [26-29].

2.2 The Need for Support

Appropriate end-user support has for a long time been recognised as an important aspect of IS implementation and use. "The quality and cost-effectiveness with which an organisation delivers IT support to end-users can be expected to influence the productivity of the workforce" [11 p. 42].

Bergeron et al. [7] investigated different aspects of support and found that among others, the satisfaction function was highly correlated to the distance from the support; that is the shorter the perceived distance the better. This reinforces the findings of Rockart and Flannery [10] who claimed that if users are geographically scattered there is a need for supplementary support services. This sustains the assumption made in this paper, that remote mobile users may require additional or different support than most likely will be offered by traditional support functions. Also sufficient training and documentation are important factors in successful end-user support, see for instance Mirani and King [9], Shaw et al. [11], Torkzadeh and Doll [13].

As a consequence of the importance of appropriate support and the reduction of possibilities to communicate and support, created by the geographical distribution of the end-users, there exist a need for guidance on how to overcome these obstacles in design of information systems.

3 The Applied Research Approach

Design science research includes the building, or design, of an artefact as well as the evaluation of its use and performance [30]. Research frameworks of design research typically include activities of theory building, solution technology invention and evaluation, which can be natural or artificial. The research framework and the connections between the research activities are presented in figure 1. The arrows show that the researcher can, over time, alternate between different activities as the research aim dictates [31].

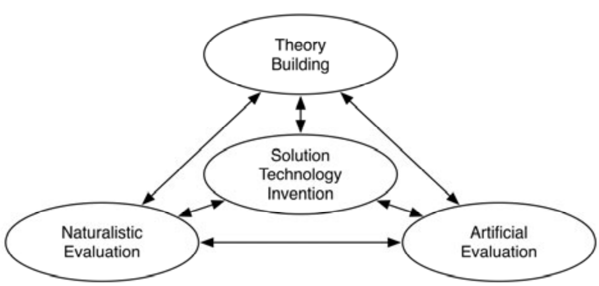


Fig. 1. The applied research framework. Adapted from Venable [31].

The study can be mapped according to Venable's [31] framework. Theory Building (problem theories) initiating the research are theories on the importance of support, a phenomena discussed by, among others, Doll and Torkzadeh [8], Rondeau et al. [32]

and Shaw et al. [11, 12]. The implemented solution at the County Council of Kalmar is part of the Artificial Evaluation. The Naturalistic Evaluation, in the form of a case study on the implemented information system, informs the researcher and is the foundation for a Solution/Technology Invention as a design of an artefact and a design proposition. Building on this knowledge, a design theory is developed, thus informing the Theory Building.

The approach used for identifying problems, a case study, was a mix of several methods and techniques as recommended by Hevner and Chatterjee [33]. A presentation of the system was attended by the author, focussing on the development of the system, and notes were taken during the presentation. Five semi-structured follow-up interviews with the developers of the WinHast system with a duration of 60-240 minutes each were held, and field notes were taken. One of the interviewed developers also worked with support to end-users as drivers and dispatch staff. Studies of online material describing the system were conducted with a focus on functionality of WinHast. After each interview, the interview outline was modified with more specific questions making it possible to dig deeper with each interview. The actual outline was in the shape of a mind map [34] and using this limited the rigidity that might come with a list of predefined questions. Complementary information was gathered by telephone interviews with the managers of the system to verify the developer's statements on perceived benefits. The parts of the implemented solution that managed support related aspects were studied closer. In the analysis, decisions and assumptions regarding design considerations to enhance the support to a remote and geographically scattered workforce were in focus.

Even if all developers at Aspea Mobile were interviewed some concerns on the number on informants can be raised. In order to manage this concern developers from another company was interviewed and design considerations in order to manage the geographical scattered workforce was among the topics. These developers belonged to IBM Mobile Payment developer team with experience from development of mobile applications. 3 developers were interviewed for about 60 minutes each and the interviews were recorded. The results from these interviews were mapped onto the previous results and analysis if the IBM developers opinion supported the strategies that were implemented in WinHast or not.

There are different opinions on how the output of design science can be expressed. The term "design proposition" is a term used foremost in management research that follows the logic of a technological rule. In the field of IS it may be more appropriate to use the term "design proposition" instead of technological rule since the latter term may suggest a technical, rather mechanistic approach. A design proposition can be expressed as: if you want to achieve X in situation Y, then something like action Z may help. The contextual dependency and the condition that design propositions must be interpreted in a specific setting also indicates that design proposition is a more suitable label than technological rule [35, 36].

4 Results from the Case Study

Aspea System, a software development firm, built the WinHast system during 2002-2009. The developers worked closely with the customer (Kalmar County Council) and the end-users (the drivers and dispatch staff) using an iterative approach. The perceived usability of the system was enhanced due to a high degree of user involvement. Several aspects on how to manage properties of mobility were elaborated on and tested, such as different interfaces on the HCD, different work flows and different hardware.

One important factor in this case is that the developers had experience from other developmental efforts, both from a mainframe environment and client-server environment. They also had experience in the increased complexity regarding support issues moving from mainframes to client-servers. That is, it was perceived as a more complex task to administer the distributed set of computers, as the PC compared to the non-geographically distributed mainframe. In the mainframe environment support was made on just one computer, whereas client server support was given to a wide range of computers, often with individual settings and far more disperse over offices and buildings forcing the support staff to travel to the different end-users. By this, demanding a larger amount of support staff and support staff with a broader knowledge base (compared to the main frame environment).

With this in mind the developers set off to design and develop a system that would minimise the support required and improve the possibilities to offer adequate support when needed. This resulted in a system with extensive monitoring features and extensive remote management or as one of the designers puts it: "we sometimes joke with the customer and say that we have a fascistoid control system – everything is monitored" (Richard Nicklasson, Aspea Mobile).

At the time of writing 210 vehicles are equipped with HCD. The HCD is a Mio A710 smart phone with Windows Mobile 5.0, and approximately 3000 driving assignments per day are handled. The application is built of three modules: the actual WinHast-application, a GPS-module and a GPRS module. An additional application bought from a third party supplier is installed for remote management.

4.1 About Synchronisation and Logs

Developers considered at first the use of Windows Sync, but it was not selected due to extensive data transfer and high data transfer costs. They tested Windows Sync and found that a regular work day (12 hours) would require 10MB worth of data transfer. In 2002 the data transfer was paid per kB and the cost for data transfer using Windows Sync had been too high in 2002. The system built used custom-developed XML-synchronisation, where one month's worth of data traffic with synchronisation every minute, never exceeded 50MB (per month) of data traffic. Another benefit from using XML- synchronisation was reduced risk of problems related to unreliable networks, that is, the lesser amount of data to transfer the lesser possibilities for problems. An additional reason to avoid Windows Sync was the encapsulated data. There were problems to "go inside" problematic synchronisations to debug them. Instead the XML data was sent without encryption within an APN-network provided by Telenor (a Swedish carrier). Proceeding in this way it is easy to study the synchronisations and to create extensive logs.

These logs are worth an section of its own in this paper, the system records and logs everything that has happened on the handheld device. It includes buttons pushed, driving speed, number of satellites in range, the GPRS signal strength, battery status, battery charging history, the position of the handheld and all driving assignments, and this information is updated every minute. From this the support staff can remind a user if the handheld is about to run out of battery power or if the user has pushed the wrong buttons when trying to achieve something that went wrong. The logs are also valuable because it is not always the case that the user calls immediately when something goes wrong, they may call a few days later, and in order to pin down exactly when and what went wrong the information in the logs is valuable.

4.2 About Fault Tolerance

The application is built to be tolerant to faults and continuously monitors the GPS and GPRS signals. If the GPS-module loses connection it restarts by itself and the same procedure for the GPRS-module. If both of these soft restarts do not do the job, the handheld performs a "coldstart", in that it closes down and does a complete restart. This functionality has saved the support and end-users a lot of time.

The handheld is used in the field during driving, and this requires that the HCD can be handled with one hand (the other hand is occupied with steering) this in combination with driving makes the environment anything but calm and stable, the car can shake and jerk making touch screen with small icons unreliable and prone to input error. These two aspects rendered a system that is manoeuvred by only two buttons (proper buttons, not on a screen) which according to the developers was delicate and difficult to achieve.

The HCD is configured as a 1-function device, this to reduce sources of problems. The only function is the driving assignments, no telephone ability and no web surfing. This is achieved by the installation of a remote management application that also allows the support to restart and or install applications remotely.

4.3 About Server Side Connections

When a HCD is started it requests information from a connection point in order to update information. It automatically polls for updates and information exchanges. If the preferred connection point is unavailable it automatically tries another connection point (there are 8 connection points, i.e. services running) until a connection is established.

The above mentioned connection points are also used as load distributors in the case of a total carrier network failure. If all mobile HCD lose connection with the GPRS network due to a carrier problem, there is a risk of overload when the network becomes available again. If all of the HCD try to connect to the same connection point it will most likely be overloaded. To avoid this, the HCD tries to connect in a preset process, segmenting and balancing the requests.

4.4 The Support Staff Situation

According to the support staff 70-80% of all support errands are solved by a remote restart. The remaining errands are instructing new drivers (which usually requires 3-4 telephone calls) and hardware support such as replacing charger cables, batteries, SIM-cards et cetera. Other tasks can consist of managing problems originating from the carriers' service. Sometimes the user does a hard reset on the handheld by mistake and support has to reinstall everything in-house. It should be noted that a single employee manages all the support and the administration work of carrier agreements and also provides support for the back office staff. An amount of time is also spent on assisting the developer in further development of the WinHast system.

4.5 Limitations of the Study

Some limitations in the study are identified, such as the evaluation of the implemented features. In this case the workload and types of support errand do function as measurements or indicators of the success of the features implemented. However that all of the developers of the WinHast system was interviewed the amount of interviews is low. In order to reduce this weakness a validation by interviewing another developer team on the subject was made.

5 Lesson Learned from the Case: Defensive Design

The developers have continuously developed the WinHast system for seven years. One beacon guiding the developers (among others) was their experience of support issues, and the assumption that support to digital rangers could be problematic if not already managed properly within the design of the system. To manage or reduce this assumed problem some design choices were made and implemented.

One can make the observation that some of the installed features are not support solutions, as for example an installed helpdesk-application, self-service application or some intuitive training facility to increase the perceived value of support. Instead it is about minimising the need for support, and specifically the need for physical on-site support performed by staff. By applying these strategies the developer have reduced some of the complexity and problems related to the geographical distance between the user and the location of support staff. Formulated as the design proposition Defensive Design the proposition is constituted by two strategies, one on preventing breakdowns and one on enhanced recovery features as illustrated in the following sections.

The structure of the following presentation is; first the recommendation, followed by an argument supporting the recommendation, finally a description of how the recommendation was managed in WinHast.

5.1 On Preventing Breakdowns

Preferred communication with low amounts of transferred data instead of more complex variants: With less data to transfer less time is required for data transfer problems. In the Winhast system, XML schemes were used for communication. Built in synchronisation features were not selected due to large amounts of transferred data.

Open synchronisation with standard protocols: By this gaining transmission easy to debug, if problems with security exits an APN network may solve the security issues. In Winhast in-house developed clear text messages in XML were used for communication and an APN network was used for telecommunication.

Frequent synchronisation and storage of log files: By this gaining continually monitoring making it possible to debug afterwards. In WinHast the HCD synchronised every minute and all the data from synchronisation was stored.

Designed for driving situation: Gaining an application that was easy to learn and use in the field with a reduced amount of improper input. In WinHast the input was performed with only a two-button interface, the touch screen was not selected as an input option.

5.2 On Recovery

Continuous process monitoring and modules with self-restarting ability: Gaining a robust application that can restart services when needed, without the need of user input. In WinHast the GPS module, the GPRS module and the HCD itself had features that performed a self-restart if needed.

Installation and use of an application for remote management: Gaining reduced need for on-site maintenance. In WinHast an application with features such as remote installation, remote updates, remote restarts and surveillance of the HCD was installed.

Segmented server side connections and load balancing between segments: Gaining load balancing in the case of network breakdowns and several alternative connection points offering a fault tolerant service. In WinHast the HCD polled after a preset schedule different connection points if the standard connection point was out of order.

5.3 Some Thoughts on Validation of Defensive Design

To validate a design proposition can be problematic due to lack of established and measurable "measurements". In this paper the low workload on the support staff (see section 4.4) is one indication of success. To go one step further, the design considerations implemented in WinHast was discussed with a mobile application developer team at IBM Denmark. The developers' recognised three different strategies or design considerations, self-recovery features as those implemented in WinHast, low complexity applications and using development frameworks as IBM Portal. However, the developer supported the implemented solutions in WinHast and the design proposition.

6 Evaluation of the Design Proposition of Defensive Design

When engaged in building artefacts, we are engaged in design [14]. The rationale for developing design propositions is the possibility to later further enhance them to design theory. The ambition is to enhance the body of knowledge for the design and development of applications for mobile users. A design theory is suggested and is evaluated according to Gregor and Jones' framework for information systems (IS) design research [37].

As the theoretical framework for portraying the properties of the suggested design theory states, Gregor and Jones' [37] work on IS design science theory is applied. According to Gregor and Jones, the first six components of the design theory are sufficient to give an idea of an artefact that could be constructed: (1) purpose and scope, (2) the constructs, (3) the principles of form and function, (4) the artefact mutability, (5) testable propositions and (6) justificatory knowledge (in this particular case the additional components of the framework were not considered relevant). The first five components have direct parallels to components proposed as mandatory for natural science theories [31, 36]. The sixth component has been added to provide an explanation to why the design works or not. Purpose and scope says "what the system is for". To understand an artefact it is necessary to understand the context and the circumstances in which it operate. To make a valid description of purpose and scope, the context and reason for the existence of the design theory must be clarified. Constructs concern representations of the entities central to the design theory, they can be assembled from words, diagrams or mathematical symbols. Principles of form and function describe how the artefact is constructed, a blueprint of the artefact. Artefact mutability illustrates the evolutionary properties of IS artefacts, that it is difficult to define a design due to this ever-changing material. An ambition could be to consider these evolutionary properties in a design theory. Testable propositions are statements of causality, either algorithmic propositions that can be tested or heuristic propositions with a form as "a likely outcome". These testable propositions are difficult due to the nature of IS, but there should be an ongoing effort to achieve these type of propositions. Justificatory knowledge concerns the explanatory knowledge that links goals and materials (see table 2).

Table 2. Six components of a Information Systems Design Theory [37]

Component	Description
Core components	
Purpose and scope (the causa finalis)	"What the system is for" - the set of meta-requirements or goals that specifies the type of artefact to which the theory applies, and in conjunction also defines the scope, or boundaries, of the theory.
Constructs (the causa materialis)	Representations of the entities of interest in the theory.
Principle of form and function (the causa formalis)	The abstract "blueprint" or architecture that describes an IS artefact, either product or method/intervention.
Artefact mutability	The changes in state of the artefact anticipated in the theory, that is what degree of artefact change is encompassed by the theory.
Testable propositions	Truth statements about the design theory.
Justificatory knowledge	The underlying knowledge or theory from the natural or social or design sciences that gives a basis and explanation for the design (kernel theories).

Following van Aken's [36] advice in formulating a design proposition, "if you want to achieve X in situation Y, then something like action Z will help" we put forward this proposition: If you want to reduce the need for on-site support to the mobile workforce (X) where the users are geographically scattered (Y) then enforce defensive design (Z). The proposed design theory for managing the off-task property of mobile information systems using Gregor and Jones' framework [37] is summarised in table 3.

Table 3. Six components of an Information Systems Design Theory for the management of support to digital rangers labelled as defensive design

Component	Description
Core Components	
Purpose and scope (the causa finalis)	The aim is to develop a system with functionality to reduce the need of support to the geographically distributed mobile workforce.
Constructs (the causa materialis)	Digital rangers, defensive design, mobile workforce, handheld computing devices.
Principle of form and function (the causa formalis)	Management of remote handheld computing devices and support issues by recovery functions and preventing breakdowns.
Artefact mutability	Suggestions for improvement during the development phase were given from the users due to a high degree of end-user involvement during the seven years of iterative development. The amount of workload on the support staff acts as an indicator of success. Another external developer team validates these suggestions.
Testable propositions	If you want to limit the problems originating from the geographical distance between remote user and support staff then implement defensive design in the information system and the handheld computing devices.
Justificatory knowledge	The underlying perspectives stem from design science and empirical studies of the properties of mobile workforce, such as the property of being digital ranger.

7 Conclusion

In order to harnessing the possibilities of mobile computing it is important to realise the nature of the mobile workforce. However the concept of the mobile workforce is ambiguous and can convey very different meanings to different readers. In this paper the concept of a digital ranger is used to depict a specific type of user class with specific needs of the computerised information system and thereto-related activities as support. Being more or less completely detached from the home base (as the digital ranger is) causes problems related to the reduced ability to physical meet with support staff, lack of supporting colleagues and the absence of supporting technologies. This circumstance may make it worthwhile to undertake some design considerations. In this paper the proposition of defensive design is put forward to reduce the problems related to the geographical distance and to support out in the field.

The main principles of defensive design are to prevent breakdowns and to enable recovery. The design theory originates from the lessons learnt by developers during the development of WinHast, a system for the administration of transportation.

It is notable that the proposed design considerations are not some kind of support solution, as for example an installed helpdesk-application, self-service application or similar. Instead it is about minimising the need for support, and specifically the need for physical on-site support performed by support staff and this is achieved by extensive prevention of breakdowns such as; minimised data transfer, easily debugged transfer protocols, frequent synchronisation and monitoring, design for the actual use situation, and recovery features if the system does fail such as; self-restarting applications, remotely managed HCDs and segmented server connections. By this the developers have reduced some of the problems related to the geographical distance between the user and the location of support staff.

One mean to evaluate the developer's opinion on the success of the implemented system can be the amount of time spent on support. An estimated 70-80% of all errands were solved via remote management, and mainly by restarts. The remaining errands were mostly problems related to hardware such as new batteries, et cetera. Taking into consideration that the support was offered by only one person (one full-time member of staff) to 210 users in the field gives an indication on the moderate level of support required. The same person also administered the carrier agreement, training for new users and hardware support, and participated in the continuous development of the WinHast system. So, less than one full time employee performed the support for these 210 HCD and 3000 driving assignments per day.

7.1 Future Work

Future work on the proposal is of course required. It would be interesting to find out if the suggestions are precise enough that they can be understood by other developers, and that they are general enough to be applicable in different settings.

References

1. Barnes, S.: Enterprise mobility: concept and examples. Int. J. Mob. Commun. 1, 341–359 (2003)
2. McIntosh, J., Baron, J.: Mobile commerce's impact on today's workforce: issues, impacts and implications. Int. J. Mob. Commun. 3, 99–113 (2005)
3. Maunuksela, A., Nieminen, M.: Micromobility supported WLAN networks: an empirical study of new IP protocol to support mobility and connection handovers. Int. J. Mob. Commun. 3, 127–137 (2005)
4. Satyanarayanan, M.: Fundamental challenges in mobile computing. In: Proceedings of the Fifteenth Annual ACM Symposium on Principles of Distributed Computing. ACM, Philadelphia (1996)
5. Shin, D.: Ubiquitous Computing Acceptance Model: end user concern about security, privacy and risk. Int. J. Mob. Commun. 8, 169–186 (2010)
6. Tarasewich, P., Nickerson, R.C., Warkentin, M.: Issues in Mobile E-Commerce. Communications of the Association for Information Systems 8 (2002)
7. Bergeron, F., Rivard, S., De Serre, L.: Investigating the support role of the information center. MIS Quart. 14, 247–260 (1990)
8. Doll, W., Torkzadeh, G.: The measurement of end-user computing satisfaction. MIS Quart. 12, 259–274 (1988)
9. Mirani, R., King, W.: The development of a measure for end-user computing support. Descision Sci. 25, 481–498 (1994)
10. Rockart, J., Flannery, L.: The management of end user computing. Commun. ACM 26, 784 (1983)
11. Shaw, N., DeLone, W., Niederman, F.: Sources of dissatisfaction in end-user support: an empirical study. ACM SIGMIS Database 33, 41–56 (2002)

12. Shaw, N., Lee-Partridge, J., Ang, J.: Understanding the Hidden Dissatisfaction of Users Toward End User Computing. Advanced Topics in End User Computing 202 (2004)
13. Torkzadeh, G., Doll, W.: The place and value of documentation in end-user computing. Inform. Manage. 24, 147–158 (1993)
14. Simon, H.A.: The Science of the Artificial. MIT Press, Cambridge (1969)
15. Fällman, D.: In Romance with the Materials of Mobile Interaction: A Phenomenological Approach to the Design of Mobile Information Technology. Informatics, p. 412. Umea University, Umeå (2003)
16. Zheng, W., Yuan, Y.: Identifying the differences between stationary office support and mobile work support: a conceptual framework. Int. J. Mob. Commun. 5, 107–123 (2007)
17. Duchamp, D.: Issues in wireless mobile computing. In: Proceedings of the Workstation Operating Systems, Key Biscayne, FL, USA, pp. 2–10 (1992)
18. Nichols, J., Myers, B.A.: Controlling Home and Office Appliances with Smart Phones. IEEE Pervas. Comput. 5, 60–67 (2006)
19. Toye, E., Sharp, R., Madhavapeddy, A., Scott, D.: Using smart phones to access site-specific services. IEEE Pervas. Comput., 60–66 (2005)
20. Andersson, B.: Mobile Computing from a Developer's Perspective: A 10-Year Review, 1999–2008. In: Forbrig, P., Günther, H. (eds.) BIR 2010. Lecture Notes in Business Information Processing, vol. 64, pp. 220–233. Springer, Heidelberg (2010)
21. Andreou, A., Leonidou, C., Chrysostomou, C., Pitsillides, A., Samaras, G., Schizas, C.: Key issues for the design and development of mobile commerce services and applications. Int. J. Mob. Commun. 3, 303–323 (2005)
22. Eppinger, S., Ulrich, K.: Product design and development. McGraw-Hill, New York (2003)
23. Dahlbom, B., Ljungberg, E.: Mobile informatics. Scand. J. Inform. Syst. 10, 227–234 (1998)
24. Orr, J.E.: Talking about machines: an ethnography of a modern job. ILR Press, Ithaca (1996)
25. Brown, B., Kenton O'Hara, B.: Place as a practical concern of mobile workers. Environ. Planning A 35, 1565–1588 (2003)
26. Perry, M., O'hara, K., Sellen, A., Brown, B., Harper, R.: Dealing with mobility: understanding access anytime, anywhere. ACM Trans. Comput.-Hum. Interact. 8, 323–347 (2001)
27. Andersson, B.: About Appropriation of Mobile Applications - The Applicability of Structural Features and Spirit. Proceedings of the European Conference on Information Systems (ECIS), Galway (2008)
28. Andersson, B., Carlsson, S.A.: Designing for Digital Nomads: Managing the High Reliance on Single Application. In: Proceedings of the Global Mobility Roundtable, Cairo (2009)
29. B'Far, R.: Mobile computing principles: designing and developing mobile applications with UML and XML. Cambridge University Press, New York (2005)
30. Baskerville, R., Pries-Heje, J., Venable, J.: Soft design science research: Extending the boundaries of evaluation in design science research. In: Proceedings of DESRIST the 2nd International Conference on Design Science Research in Information Systems and Technology, Pasadena, CA (2007)
31. Venable, J.: A framework for design science research activities. In: Proceedings of the Information Resource Management Conference, Washington D.C. (2006)

32. Rondeau, P., Ragu-Nathan, T., Vonderembse, M.: How involvement, IS management effectiveness, and end-user computing impact IS performance in manufacturing firms. Inform. Manage. 43, 93–107 (2006)
33. Hevner, A., Chatterjee, S.: Design Science Research in Information Systems: Theory & Practice. Springer, Berlin (2010)
34. Buzan, T., Buzan, B.: The mind map book. Pearson Education, London (2006)
35. Bunge, M.: Technology as applied science. Tech. Cult. 7, 329–347 (1966)
36. Van Aken, J.: Management research as a design science: articulating the research products of mode 2 knowledge production in management. Brit. J. Manage. 16, 19–36 (2005)
37. Gregor, S., Jones, D.: The anatomy of a design theory. J. Assoc. Inf. Syst. 8, 312–335 (2007)

Design Range and Research Strategies in Design Science Publications

Philipp Offermann, Sören Blom, Marten Schönherr, and Udo Bub

Deutsche Telekom Laboratories, Ernst-Reuter-Platz 7, 10587 Berlin, Germany
{philipp.offermann,soeren.blom,
marten.schoenherr,udo.bub}@telekom.de

Abstract. Not much is known about how design knowledge is re-used in Design Science Research (DSR). The concepts of "generalization" and "transfer" as different types of knowledge-building and re-use are discussed in other disciplines and in Information System Research in general, but less so in DSR. Offermann et al. [1] proposed three ranges of design theories and seven strategies for how to create and generalize design knowledge. In this paper, we classify all DESRIST 2006 – 2010 publications according to design range and research strategy. By doing so, we empirically ground the merely theoretically established research strategies, and are able to discover three additional strategy types. The literature analysis shows that the specification of design range and strategy in abstracts is often incomplete or misleading. Based on the analysis we recommend template abstracts for design science publications which guide researchers on how to include all relevant information about design knowledge they (re-)used.

1 Introduction

In design science research, different design ranges and research strategies are available to generate and structure research output. Offermann et al. [1] propose three ranges for design theories. Short-range designs present a solution to a specific problem an entity (e.g. company, government department) has, thereby only applicable to the specific setting the design has been developed for. Mid-range designs are more general in that they do not contain any information specific to a single case, but present a more generic description of a solution to a generic problem. Long-range designs are generalized principles extracted from a set of mid-range designs, informing a range of related activities and presenting general insights about a type of design approach. Offermann et al. [1] present seven strategies for how to create and generalize design knowledge, which use the three ranges to characterize input and outputs.

The aim of this paper is to improve the comprehensiveness of the design and research strategy categorization and resolve incompletion and problems. A solid and usable categorization scheme can enable design science scholars to reflect and communicate about their methods and approaches across different functional domains and artifact types. More importantly, it can help to make the re-use of design knowledge more efficient. The "producer" of knowledge has a reference against which the

H. Jain, A.P. Sinha, and P. Vitharana (Eds.): DESRIST 2011, LNCS 6629, pp. 77–91, 2011.

work can be described and the "consumer" of knowledge can describe the information need in a more standardized way which promises better matches in literature search and during evaluation of potentially related work.

To achieve this, we analyzed the DESRIST conference proceedings from 2006 through 2010. For each publication presenting a design, we classified which design range the discussed design belongs to and the research strategy used in the paper. For publications that we were unable to classify, we discussed if a design range or a research strategy was missing from the Offermann et al. [1] publication. To support the applicability of our findings, we decided to propose abstract templates with which the design work in a paper can be described in a standardized way according to the strategies and design ranges.

The paper is structured as follows: The three types of design ranges and seven research strategies presented in Offermann et al. [1] are introduced in more detail first, followed by an explanation of our research methodology and the literature analysis process. The results of the analysis and the identification of missing ranges and designs are presented next. Finally, the results and their implications are discussed and a conclusion is drawn.

2 State of the Field

Hevner et al. [2] describe IS research as the interaction between the (business) environment and the ISR knowledge base. The research has to be applicable in the appropriate environment, and at the same time provide additions to the knowledge base. This knowledge base is used to generate new designs by abduction [3]. As "knowledge becomes 'relevant' when it is context specific" [4] to fulfill business needs, an artifact designed needs to be as specific as possible in respect to people, organizations and technology. The more adapted a design is to a specific setting in practice the more relevant it is, as instantiations are easier to generate. On the other hand, the more specific a design is the narrower the scope, and the less likely to find a case for another instantiation.

There seems to be some awareness of the relevance of the level of abstraction in the community. However, generalization and transfer have not received much concentrated attention. The most explicit statements about generalization that we could find were: "The design scientist must be able both to generalize the findings and demonstrate a theoretical contribution." [5] and "Design-science research holds the potential for three types of research contributions based on the novelty, generality, and significance of the designed artifact." [2].

The first step in describing the creation and re-use of design knowledge is to understand the different ranges of design. Offermann et al. [1] introduced three different "ranges" of design, based on an analogy to theory ranges from the field of sociology [6] and prior work by Holmström et al. [5]. An overview of the three design types is presented in table 2. To create new designs and transform designs from one range to another, Offermann et al. [1] identify seven research strategies presented in table 1. The identification is based on theoretical considerations by the authors. Each strategy identified is supported by a case from a scientific publication [1].

Table 1. Research strategies in Design Science

Strategy	Affected types	Approach	Research contribution	Validation
Explore new problem	Short-range	Invent design for new problem	First-of-a-kind design offers first design insights on new problem	At least one real-life instance validates utility
Validate mid-range design	Mid-range to short-range	Create new short-range design and validate its utility	Increased generalizability of utility statement	Use of accepted evaluation strategies
Generalize to mid-range design	Short-range to mid-range	Analyze commonalities and differences of short-range designs with comparable purpose and scope and find generalized representation	Captures generalized knowledge in terms of common design elements about a problem domain	The process of identifying similarities and finding generalized representation of concepts. Demonstration of applicability of new mid-range design by creating a new short-range design from it
Apply out of scope	Mid-range to short-range	Derive short range design from mid-range design and change it to work for new problem	Indication that mid-range design might cover wider scope and possibly first-of-a-kind design	At least one real-life instance validates utility in a setting outside of the original scope
Synthesize mid-range design	Mid-range to mid-range	Analyse commonalities and differences of mid-range designs with comparable purpose and scope and find generalized representation	Make mid-range design better transferable and possibly increase utility	The process of identifying similarities and finding generalized representation of concepts. Demonstration of applicability of new mid-range design by creating a new short-range design from it
Combine designs	Mid-range to mid-range	Merge designs with adjacent purpose and overlapping scope	Create a design with a more comprehensive purpose	The process of combining the designs. Demonstration of applicability of new mid-range design by creating a new short-range design from it
Extract long-range design	Mid-range to long-range	Analyze commonalities and differences of mid-range designs from the same domain and identify common principles	Captures design principles that apply to a whole class of problems	The process of identifying the design principles

Table 2. Types of design according to range of scope

Design type	Definition	Role in design	Role in research	Examples
Short-range design	Design for a specific setting	An instance (system implementation, method enactment) can directly be derived from the design	First-of-a-kind solution to a relevant problem	The specification for a CRM system; the software development process for a company
Mid-range design	Design for a specific type of setting	The design can be used to create a short-range design for a particular solution of the same problem domain	Identification of relevant design elements for a particular problem domain	eXtreme Programming, TOGAF, Rational Unified Process, relational database design
Long-range design	General insights about a type of design approach	Educational, as a starting point for dealing with a problem, illustrating a particular design "world-view"	Inform more specific designs	SOA, Object-Orientation, relational data-management, agile software development

3 Literature Review

The research in this paper is a qualitative literature review. We use a methodology similar to Offermann et al. [7]. It is based on the recommendations from Webster and Watson [8], the methodologies described by Creswell [9] and the example paper from Stelzer [10]. Table 3 gives an overview of this approach. The remainder of this section describes each step in more detail.

Table 3. Overview of literature review approach

Activity	Description	Results
Data set definition	Determine the set of articles to review, exclude those without abstracts	Set of 148 articles, 145 with abstract
Filtering of design articles	Identify the articles that contribute a design, based on title and abstract	Subset of 70 articles
Individual categorization	Each researcher: identify range of artifact and strategy of artifact creation, according to existing categories. Take note of any potentially new strategy candidate	Three individual classifications and list of new strategy candidates
Unification and description of new strategies	For each newly identified strategy: Offer definition and discuss one design example, similar to [7]	Definition and design example for each of the five new strategies
Consolidation of categorization	Discuss those cases where dissenting categorization has taken place. Try to resolve different interpretations of design and / or strategy. Discuss fit of new strategy candidates	Unified classification using old and new strategies

3.1 Data Set Definition

The data set contains 148 papers from the DESRIST proceedings of the years 2006 through 2010. We used title and abstract of each paper for the analysis only. Consequently, three papers without abstract were excluded. Three authors of this paper were involved in the literature review and interpretation; the research was conducted between December 13th, 2010 and January 3rd, 2011.

3.2 Filtering of Design Articles

To determine which articles were of further interest, we had to decide which papers to classify as design science, and include in the review, and which to exclude. We decided to include papers presenting prescription-driven design science according to van Aken [11] and papers prescribing design and action according to Gregor [12].

Based on these two papers, we developed two questions to include and exclude publications: *Who uses the results?*, with *practitioners* being in scope and *researchers only* being out of scope; and *How are the results used?* with *to guide action* being in scope and *to understand the world, to inform only* being out of scope. These are the same criteria as used in the previous survey of DESRIST design papers in [7].

Each of the researchers looked at each of the 145 papers as to whether it contained a designed artifact. In 62 cases we independently classified a paper as not presenting a designed artifact and 54 we independently identified as designs. Of the 39 papers with dissenting assessments, a consensus could be reached after a discussion, excluding four cases. After the discussion, 70 papers were considered to be a design and thus used in the further steps.

Table 4. Amounts of short-, mid- and long-range designs identified independently by each researcher

Category	Reviewer 1	Reviewer 2	Reviewer 3
Short-range design	21	13	22
Mid-range design	48	55	47
Long-range design	1	2	1

3.3 Individual Categorization of Strategies and New Strategy Candidates

The individual categorization consisted of three steps: The researchers first had to determine the design range (short, mid, or long). They then had to try to find a suitable strategy out of the strategy catalog defined in [1]. If that was not possible, the researcher had to come up with a new strategy that captures the approach of the paper. Table 4 shows the results of the three individual classification of designs into short-, mid-, and long-range. The largest differences were the classification of some designs into either mid- or short-range.

Fig. 1 shows the results of the individual classification, aggregated by unanimous and majority classification. The first seven bars show the seven previously defined strategies, the eighth bar shows the number of papers in which new strategies were proposed (not necessarily the same strategies yet) and the ninth bar represents the number of completely diverting classification, in which each research offer a different opinion. The dark segments of each bar show how many papers were classified by all three researchers in the same way, whereas the light segments represent those classifications where two of the three researchers were of the same opinion.

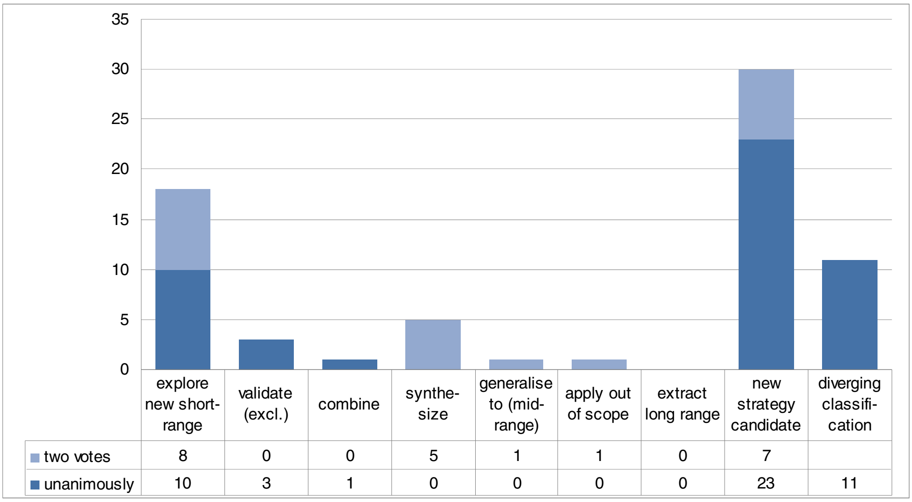


Fig. 1. Numbers of unanimous and majority classification for the existing seven strategies, new strategy candidates and diverging classification

Table 5. Candidates for new design strategies

Reviewer 1	Reviewer 2	Reviewer 3
Improve new short-range		Improve short-range
		Extend short-range
	Short-range out of scope	
Explore new mid-range	Propose new mid-range	Explore new mid-range
Improve mid-range	Improve mid-range	Improve mid-range
Increase scope (mid-range)	Extend scope (mid-range)	Extend mid-range
	Specialize mid-range	
	Propose alternative (mid range)	
Deduct mid-range from long-range		
Infer new long-range from theoretical considerations		

Finally, new strategies emerged whenever it became clear that no existing strategy would fit without bending the definition. Each researcher named and preliminarily defined their new candidates, which are shown in table 5.

3.4 Description of Newly Discovered Strategies

We discussed each proposal in turn to select a final set of strategies out of the candidate list. For those that appeared on each candidate list, not much discussion was necessary beyond ensuring that everyone had a compatible definition in mind. For those with one or two mentions we examined the papers the researcher considered to exemplify the strategy. In some cases (e.g. “extend short-range” and “short-range out of scope”) it turned out that the strategies applied, but that the design was in a different range (i.e. mid-range). In other cases (e.g. “propose alternative (mid-range)”) a solid distinction from other strategies (“improve mid-range” in this case) was theoretically possible but did not manifest clearly enough in the abstract. Finally, we identified three new strategy types and two additional variations (cf. fig. 2). The individual strategies are differentiated by the following criteria:

- Design range(s) affected: Short-, mid-, long-range design.
- Multiplicity of designs used as input: One design as input design vs. multiple designs as input.
- Change of purpose/scope: Purpose and scope remains constant vs. purpose is changed and/or scope increased.
- Increase of utility: Utility remains constant vs. utility is increased.

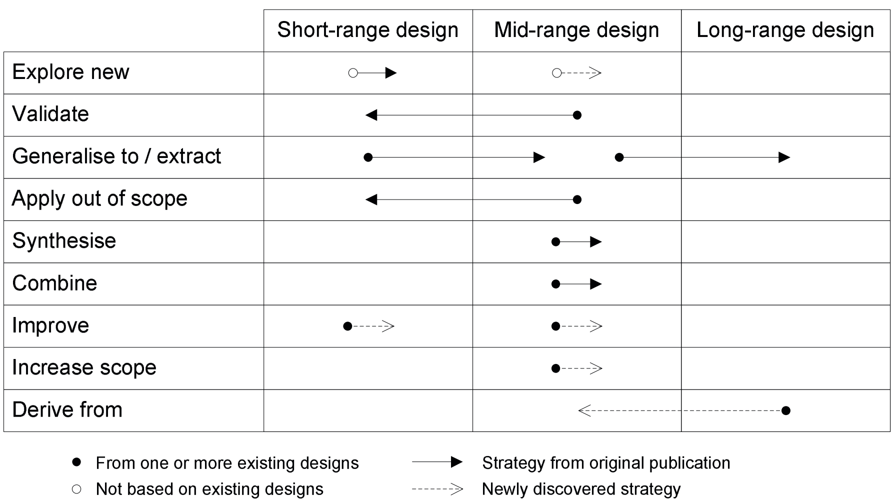


Fig. 2. Extended set of research strategies

In the original set of strategies, a new design could only be invented as short-range. In the literature analyzed, however, we found many research results that explore a new design not as short-range, but as mid-range. Contrary to other strategies involving mid-range designs, *explore new mid-range* does not build on prior short-range or mid-range designs. Usually when employing this strategy, a validation is performed at the same time to demonstrate the utility of the newly developed design.

We also found papers that propose improvements to existing designs in order to create a better design. This happens for short- and mid-range designs: *improve short-range design* and *improve mid-range design*. The improvements concern the design's utility compared to existing designs, keeping purpose and scope constant.

Another possibility we found was to increase the scope of an existing mid-range design. *Increase scope* keeps the purpose of a design constant but makes it applicable in more cases. The strategy is similar to "combine designs", with the difference that only one existing mid-range design is involved and the extensions are newly developed.

Finally, based on an existing long-range design, one can *derive* a new mid-range design that realizes the ideas from the long-range design. The long-range design is operationalized to inform practice.

To highlight the newly discovered strategies, we present examples of DESRIST papers using each of the strategies. In his paper "A Design Language for Developing and Simulating Implementation Strategies for Knowledge Management Systems (KMS) in Small to Mid-size Enterprises (SME)", Judge [13] describes the exploration of a new language/notation mid-range design. In the abstracts, he writes: "Although there has been extensive research performed on aspects of these issues, and some attempts to model Knowledge Management, to the best of my knowledge no one has developed a design language to create simulations specifically for understanding the flow of knowledge in a given organization." [13] That shows that the language he proposes is not based on any existing design. Also, the language is not specific to any concrete KMS or SME and is classified as mid-range design.

Garrett et al. [14] published a paper "Extending the Elgg Social Networking System to Enhance the Campus Conversation". They extented the existing social networking software Elgg "to support student collaboration and peer learning". "The modified software represents an alternative model to the traditional course management system, using blogs & wikis to support student collaboration and peer learning." [14] Therefore, they improve the short-range design software system Elgg by modifying the source code and creating an improved short-range design.

Erenkrantz et al. [15] published a paper "Rethinking Web Services from First Principles", where they propose improvements to the REST pattern/algorithm mid-range design in form of the protocol and architecture style "Computational REST". The new mid-range design builds on REST, but improves the integration of Web Services into the web.

In their paper "Towards Deterministically Constructing Organizations Based on the Normalized Systems Approach", Van Nuffel et al. [16] increase the scope of the mid-range method design "normalized systems approach". "Recently, the normalized systems approach has been proposed to design information systems exhibiting proven evolvability. In this paper, we extend the approach's basic principles to the related fields of Enterprise Architecture (EA) and Business Process Management (BPM)." [16] Therefore, the scope is increased from information systems design to EA and BPM.

Nakakawa et al. [17] published the paper "Towards a Theory on Collaborative Decision Making in Enterprise Architecture", where they derive a new mid-range design from the long-range design "collaborative decision making". "This paper, therefore, presents an evolving theory that is currently being used to guide the development of a method for supporting collaborative decision making during enterprise architecture creation." [17] So based the long-range design, a new method of mid-range design is developed.

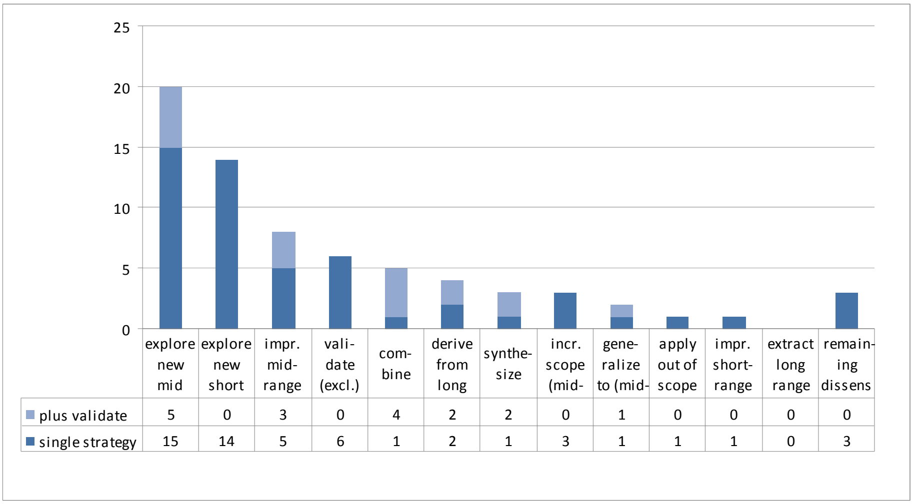


Fig. 3. Number of occurrences of old and new strategies. Accumulation of strategies without (dark segment) and with additional validation strategy (light segment).

3.5 Consolidation of Categorization and Application of New Strategies

As a final step we went through all dissenting classifications and came up with an agreement. At the same time we had to verify that they not only matched the abstracts from which they originated but also applied to new papers. In some cases, a dissenting classification was best resolved by applying a newly defined strategy.

We looked at each of the 33 papers considered to present a design but where researchers came up with different strategies. We were able to agree on a strategy in all but three cases. Two patterns reoccurred in the discussion that help to understand why we classified papers differently in the first place but were able to resolve this later.

1) Researcher focused on different "signal words" in the abstract. Some abstracts contained indicators about different strategies. If one research did not see or disregarded a signal word that the other found important, different classifications would occur. In only three cases were we not able to resolve our differences about abstracts ambiguous in this sense.

2) Signal words vs. actual mention of source designs. Some abstracts would mention words such as "synthesize" or "improve" but would not make it clear that they made use of a specific design(s) as source(s). We agreed on a shared interpretation of our own definitions, which would only consider the more specific strategy (i.e. "synthesize" over "improve" or "improve" over "explore new") if the source design was named or it was otherwise clear that specific designs were analyzed as input for the design in the paper. In all other cases, the more general strategy was used.

Fig. 3 gives an overview of the number of occurrences of joint unanimously and consensual classification (67 of 70 design papers). The “validate” strategy appears in two functions: As an addition to other strategies (e.g. a design was newly introduced or improved and then validated), or it was an exclusive strategy; in which case the paper concerned the validation of a design published in a different paper (by the same or a different author). We decided to introduce this distinction to address the fact that the papers that combined some strategy with validation were always dominated by that other strategy. The validation would not have taken place without the introduction of the design in the first place. The second type of “validate” occurrences existed due to some explicit interest in the validation.

4 Discussion

4.1 Methodological Considerations

The articles are all taken from a single conference. This might be considered one-sided. Yet DESRIST is one of the few places in which the common theme is *design* rather than some business or technical specialization. That leads us to assume that DESRIST is a good source for a wide variety of design publications, which is what we needed to check our theoretically derived strategies against reality, as well as to do exploratory research on new strategies. As we are not interested in generalizable quantitative statements, we feel that this data set is of sufficient size and quality.

We focused on abstract and title of each paper to determine the designs and strategies. By doing so, we risked missing aspects that would have become apparent inside the paper. At the same time, we considered it reasonable to assume that between abstract and title enough information communicated as to what the approach and the result of the research are. Even across disciplines, abstracts are considered to capture all relevant parts of a publication [18, 19]. Every other part of a paper might either not occur dependably or might differ widely with regards as to what is discussed in it. The other reason to focus on the abstract was that it emulated the perspective of scholars which perform literature research for their everyday work. If a paper represents a design, the fact that it is a design, what kind of design, how it was derived and what the design is based on are crucial information and they thus should appear in the abstract.

Finally, the classification was based on interpretations of three individuals. A similar study with different researches might lead to different outcomes. Generally speaking, this is possible with any qualitative and exploratory study. The fact that we were focusing on abstract and title only reduced that risk as for each article only a limited amount of text was read. The other measure was the step-wise approach of classification (is it a design? - what range? - which strategy?) and the sequence of individual classification and group discussion. Papers were classified more than once and new strategies were applied in more than one instance, whenever possible. While this does not guarantee universality, it indicates some degree of reproducibility for other researchers and other papers.

4.2 Results

The purpose of this literature analysis was to ground our theoretical proposal in actual data and to discover new strategies. For each of our seven original strategies we found several papers, except for "extract long-range". This is not surprising, given that we considered long range designs to emerge over years while becoming "text book knowledge". This type of work is usually not seen in single articles on conferences, but would probably be found in journal articles or monographs.

Even if we did not focus on the qualitative evaluation of the data, it seems noteworthy that the two most frequently found categories, covering half of all papers (34 out of 67), are new designs. We classified a paper as "new design" when the authors either claimed that a design was new or if it was not apparent that any pre-existing design had been used. In any of those cases, design researches are influenced by pre-existing design and possibly did not explicitly mention it in the abstract, so our numbers probably are too high. Even then, it suggests that knowledge re-use either does not take place or is not communicated as well as we would consider necessary. This finding is not completely new, but merely supports existing criticism. Within the Software-Engineering community, for example, the pattern that researchers declare new types of solutions has been critically discussed under the label "research-then-transfer" [20] or "advocacy-based research" [21].

4.3 Template Abstracts

Our experience with reading these many abstracts lead us to believe that researches might not make crucial information regarding their design explicit in the abstract. This reduces the usefulness of abstracts and consequently makes knowledge building complicated and less effective. As a practical contribution we suggest templates for papers presenting information system designs that attempt to fix these shortcomings. Each paper should state whether it addresses a problem for the first time or builds on previous designs. In the latter case, it should explicitly state the (class of) designs it builds on and how it uses the designs to come up with its own solution. By doing so, it also should become clear if the design addresses the original (class of) problems, or if the scope is changed or extended. Our template proposals are based on the design science artifact types identified by Offermann et al. [7], and the design range and the research strategies as presented in fig. 2. Generally, an abstract should contain a description of the purpose, the method, the results and a conclusion. [18, 19, 22] Our template abstracts cover the purpose and method aspect only, as the results and the conclusion depend mainly on the content and have to be written by the authors.

We are not the first to suggest abstract templates for systematically indicating what research is about. Newman [23] suggest so called *pro forma abstracts* for describing the results of HCI research. He identifies five types of research papers: Three kinds of *enhancements*, namely of *models* (EM), *solutions* (ES) and *tools* (ET), a *radical solution* (RS), and *experience / design heuristic* (XH). For each type a pro-forma abstract (i.e. an abstract template) is offered. The separation of different artefacts (models, solutions and tools) is more fine-grained than our classification. The differentiation between enhancement and radical solution can be found in our categorization as well (as in, for example, "improvement" vs. "propose new").

Table 6. Template abstracts

Strategy	Template abstract	Example abstract
Explore new	In the field of [field of research], the problem of [problem description] has not yet been solved. In this paper, we propose a solution to the problem in form of [artifact type]. The solution is a [short|mid-range] design [solution name] that is applicable to [scope]. (We validate the utility of the design by applying it to [evaluation setting]. [Result of evaluation])	[25]
Validate	In the field of [field of research], a mid-range design in form of [artifact type] has been proposed to solve the problem of [problem description]. In this paper, we validate the design by [evaluation]. [Result of evaluation]	[26]
Generalize to / extract	In the field of [field of research], many [short|mid-range] solutions in form of [artifact type] exist to solve the problem of [problem description]. We look at the existing designs, identify commonalities and idiosyncrasies and propose a more general [new range] design. The new design can be used to inform [enlarged scope].	[27]
Apply out of scope	In the field of [field of research], the mid-range design [existing design] is usually used to [current scope]. In this paper, we evaluate if the design can also be used for [new scope] to [new problem]. [Result of evaluation]	[28]
Synthesize	In the field of [field of research], the problem of [problem description] can be solved by different designs. Available designs are [existing designs]. All of these designs have advantages and disadvantages. We analyzed the designs and propose a synthesized design that combines the strong points of the existing designs while overcoming their weaknesses.	[29]
Combine	In the field of [field of research], the problems of [problem description 1] and [problem description 2] often occur together. The first problem can be solved by [design 1], the second problem by [design 2]. We analyzed both designs and propose a combined design with an enlarged scope that addresses both problems at the same time.	[30]
Improve	In the field of [field of research], the problem of [problem description] is usually solved by [state-of-the-art designs]. All of these (short|mid)-range designs have shortcomings, because [shortcomings]. We propose an improved design that overcomes these shortcomings by [improvements]. (We validate the utility of the design by applying it to [evaluation setting]. [Result of evaluation])	[15]
Increase scope	In the field of [field of research], the [existing mid-range design] is meant to be used to [current purpose] for [current scope]. In this paper, we propose extensions to the design so it also can be used for [new scope].	[16]
Derive from	In the field of [field of research], the idea of [long-range design] proposes solutions to [set of problems]. Based on these concepts, we developed a new solution to the problem of [problem] in form of a [artifact type] mid-range design.	[17]

Our approach is more granular in differentiating between improvements (remaining within existing scope and purpose) generalization (to mid and long range) and other strategies (changing either scope or purpose). In the realm of software engineering, Shaw examines the question “What makes good research in software engineering?” in a paper with the same title [24]. She defines three dimensions for classification: the *research question*, the *research output* and the *research validation*. For each, several instances are offered together with characteristic fragments that help to identify a particular instance. While one can use this work to build abstracts from these fragments, it does not appear to be the intention of Shaw, who rather considers this an analytical framework. Also, the work has a wider focus, including design and research in general.

5 Conclusion

We have analyzed 148 papers from the DESRIST 2006 till 2010 conferences to verify the design strategy categories presented by Offermann et al. [1]. While we were able to classify most of the papers according to these strategies, we also identified some new strategies how to create knowledge in design science. Specifically, we identified “improve”, “increase scope” and “derive”. We are confident that the enlarged set as presented in fig. 2 is more comprehensive and covers most of the viable design science research strategies.

Based on the enlarged set, we proposed template abstracts that can be used for publications. By using one of the templates as a starting point, we hope to give a clearer direction and more structure to research projects and publications. After having read all DESRIST abstracts we have the impression that a considerable number of papers could be improved by focusing on one of the proposed research strategies.

On a more general level, we hope that this paper fosters a discussion about viable research strategies in design science. We are convinced that by agreeing on a canonical set of strategies, research quality can be improved. In our study, we classified half of the design papers as “explore new”. In the light of aiming at cumulative knowledge creation, this is a worrying result. We hope that a clear specification of source and the target designs will reduce that problem.

References

1. Offermann, P., Blom, S., Bub, U.: Strategies for Creating, Generalising and Transferring Design Science Knowledge – A Methodological Discussion and Case Analysis. In: 10th Internationale Tagung Wirtschaftsinformatik, Zürich (2011)
2. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28, 75–105 (2004)
3. Vaishnavi, V., Kuechler, W.: Design Research in Information Systems (2004/2005), `http://desrist.org/design-research-in-information-systems` (retrieved January 20, 2004, last updated August 16, 2009)
4. Aram, J.D., Salipante Jr., P.F.: Bridging Scholarship in Management: Epistemological Reflections. British Journal of Management 14, 189–205 (2003)
5. Holmström, J., Ketokivi, M., Hameri, A.-P.: Bridging Practice and Theory: A Design Science Approach. Decision Sciences 10, 65–87 (2009)
6. Merton, R.K.: Social Theory and Social Structure. The Free Press, New York (1968)

7. Offermann, P., Blom, S., Schönherr, M., Bub, U.: Artifact types in information systems design science – A literature review. In: Winter, R., Zhao, J.L., Aier, S. (eds.) DESRIST 2010. LNCS, vol. 6105, pp. 77–92. Springer, Heidelberg (2010)
8. Webster, J., Watson, R.T.: Analyzing the Past to Prepare for the Future: Writing a Literature Review. MIS Quarterly 26, xiii-xxiii (2002)
9. Creswell, J.W.: Research Design - Qualitative, Quantitative, and Mixed Methods Approaches. SAGE Publications, Thousend Oaks (2009)
10. Stelzer, D.: Enterprise Architecture Principles: Literature Review and Research Directions. In: TEAR 2009, pp. 21–36 (2009)
11. 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, 219–246 (2004)
12. Gregor, S.: The Nature of Theory in Information Systems. MIS Quarterly 30, 611–642 (2006)
13. Judge, R.: A Design Language for Developing and Simulating Implementation Strategies for Knowledge Management Systems (KMS) in Small to Mid-size Enterprises (SME). In: DESRIST 2007. CGU, Pasadena (2007)
14. Garrett, N., Thoms, B., Soffer, M., Ryan, T.: Extending the Elgg Social Networking System to Enhance the Campus Conversation. In: DESRIST 2007. CGU, Pasadena (2007)
15. Erenkrantz, J., Gorlick, M., Taylor, R.N.: Rethinking Web Services from First Principles. In: DESRIST 2007. CGU, Pasadena (2007)
16. Van Nuffel, D., Huysmans, P., Bellens, D., Ven, K.: Towards Deterministically Constructing Organizations Based on the Normalized Systems Approach. In: Winter, R., Zhao, J.L., Aier, S. (eds.) DESRIST 2010. LNCS, vol. 6105, pp. 242–257. Springer, Heidelberg (2010)
17. Nakakawa, A., van Bommel, P., Proper, E.: Towards a Theory on Collaborative Decision Making in Enterprise Architecture. In: Winter, R., Zhao, J.L., Aier, S. (eds.) DESRIST 2010. LNCS, vol. 6105, pp. 538–541. Springer, Heidelberg (2010)
18. Borko, H., Chatman, S.: Criteria for Acceptable Abstracts: A Survey of Abstracters' Instructions. American Documentation 14, 149–160 (1963)
19. Weil, B.H.: Standards For Writing Abstracts. American Society for Information Science 21, 351–357 (1970)
20. Potts, C.: Software-Engineering Research Revisited. IEEE Software 10, 19–28 (1993)
21. Glass, R.L.: The Software-Research Crisis. IEEE Software 11, 42–47 (1994)
22. ISO: Documentation — Abstracts for publication and documentation. ISO 214-1976 (1976)
23. Newman, W.: A preliminary analysis of the products of HCI research, using pro forma abstracts. In: SIGCHI Conference on Human Factors in Computing Systems: Celebrating Interdependence. ACM, Boston (1994)
24. Shaw, M.: What makes good research in software engineering? International Journal on Software Tools for Technology Transfer 4, 1–7 (2002)
25. Miller, T., Leroy, G., Huang, J., Chuang, S., Charlop-Christy, M.H.: Using a Digital Library of Images for Communication: Comparison of a Card-Based System to PDA Software. In: DESRIST 2006. CGU, Claremont (2006)
26. Vahidov, R., He, X.: Design for Situated Decision Support: Application to Personal Finance. In: DESRIST 2008, Atlanta, Georgia, pp. 1–6 (2008)
27. Keller, C., Gäre, K., Edenius, M., Lindblad, S.: Designing for Complex Innovations in Health Care: Design Theory and Realist Evaluation Combined. In: DESRIST 2009. ACM, Malvern (2009)

28. Young, M.M.: Twitter Me: Using Micro-blogging to Motivate Teenagers to Exercise. In: Winter, R., Zhao, J.L., Aier, S. (eds.) DESRIST 2010. LNCS, vol. 6105, pp. 439–448. Springer, Heidelberg (2010)
29. Lahrmann, G., Marx, F.: Systematization of Maturity Model Extensions. In: Winter, R., Zhao, J.L., Aier, S. (eds.) DESRIST 2010. LNCS, vol. 6105, pp. 522–525. Springer, Heidelberg (2010)
30. Kunene, K.N., Weistroffer, H.R.: Design of a Method to Integrate Knowledge Discovery Techniques with Prior Domain Knowledge for Better Decision Support. In: DESRIST 2006. CGU, Claremont (2006)

On Expanding the Scope of Design Science in IS Research

Robert O. Briggs[1] and Gerhard Schwabe[2]

[1] Center for Collaboration Science, University of Nebraska at Omaha
Omaha, NE 68182
rbriggs@unomaha.edu
[2] Information Management Research Group, University of Zurich
Binzmühlestrasse 14, 8050 Zurich, Switzerland
schwabe@ifi.uzh.ch

Abstract. Design Science Research (DSR) has sparked a renaissance of contributions to IS, but its rigor and value of DSR could be increased by expanding its scope beyond its engineering roots to bring all modes of scientific inquiry to bear – exploratory, theoretical , experimental, and applied science / engineering (AS/E). All DSR Cycle activities can be realized as instances of one or more of the four modes. The rigor of DSR can therefore be defended in terms of the goals, research products, and standards of rigor already established for each mode. There is, moreover, a synergy among the modes that can only be realized when all four are brought to bear, because each informs the other three. To exclude any mode of inquiry from DSR, therefore, is to impoverish knowledge about its objects of inquiry. Based on these insights, we propose a modified Cycles Model for DSR realized under the disciplines of the four modes of scientific inquiry.

Keywords: Design Science, Scientific Methods.

1 On the Value of DSR

Design *Science* Research (DSR) makes important contributions to the information systems (IS) literature beyond those made by behavioral research [1]. Behavioral research focuses on the human element of IS, e.g., system usage [2], emotion in IS [3], and information overload [4]; its prevailing modes of inquiry are theoretical and experimental research. The primary mode of inquiry for DSR, by contrast, is engineering [5, 6], and DSR has as its objects of inquiry: a) Design Processes (methods and practices): e.g., agile development [7]; b) Design Products (ways of modeling IS): e.g., UML [8]; and c) Designed Artifacts (instances of technology): e.g., relational databases [9,10]. Hevner and Chatterjee [5] define three cycles for DSR: a) the Relevance Cycle for gathering requirements and field testing; b) the Design Cycle for building and evaluating design artifacts and processes; and c) the Rigor Cycle for grounding design efforts in the knowledge base and contributing

H. Jain, A.P. Sinha, and P. Vitharana (Eds.): DESRIST 2011, LNCS 6629, pp. 92–106, 2011.

knowledge to it. The formalization of DSR [1], and a DSR methodology [11] sparked a renaissance of contributions [12].

The value and rigor of DSR could be increased, however, by expanding its scope beyond its engineering roots to bring all modes of scientific inquiry to bear – exploratory, theoretical, experimental, and applied science/engineering (AS/E). All activities of the DSR Cycles [1, 5] can be realized as instances of the four modes of scientific inquiry. Indeed, a *single* DSR study could make exploratory, theoretical, experimental, and AS/E contributions. A DS researcher, for instance, having used a kernel theory to inform design choices for an IS artifact, might validate the solution by comparing it to a prior solution. If unexplained phenomena were to manifest during validation, these would contribute to *exploratory research*. If negative findings were to inspire improvements to the theory, that would contribute to *theoretical research*. If findings were positive, that would be an experimental test of the kernel theory, which would contribute to *experimental research*. If validation proved the new artifact to be superior, that would contribute to *AS/E*.

This paper argues that all DSR Cycle activities can be realized as one or more of the modes inquiry, so the rigor of DSR contributions can be defended in terms of the goals, research products, and standards of rigor for each mode. The paper demonstrates a synergy among the modes of scientific inquiry that can be tapped only by bringing all four to bear (Figure 1). It argues, therefore, that broadening the scope of DSR to incorporate all modes could increase the depth and value of DSR contributions. Based on these insights, the paper proposes a modified Cycles Model for DSR activities realized under the disciplines of the four modes inquiry (Figure 2).

2 On DSR Activities as Exploratory Research

The goals of exploratory research are to discover and describe unexplained phenomena, their correlates, and the contexts in which they manifest [13, 14]. A *phenomenon* is an outcome whose value varies across time, contexts, and conditions, for example, system reliability or user productivity. The phenomena of interest to DS researchers would be the outcomes that designed artifacts are meant to improve. In the Design Cycle, they would be embodied in design objectives for requirements and evaluation metrics for validation. C*orrelates* of a phenomenon are other phenomena whose variations appear to be related to it [15]. For example, end-user satisfaction sometimes varies with end-user user involvement in design processes (e.g., [16]).

Products of Exploratory Research: The products of exploratory research are descriptive reports of phenomena, their correlates, and the contexts where they manifest. In DSR these may be, for instance, reports of challenges in the user environment. Phenomena are generalized to explicitly defined constructs, which may be classified in taxonomies and synthesized synthesize grounded theories, which are correlative networks of interrelated constructs [13]. Grounded theories may predict outcomes in the contexts where they were developed, but may not generalize to other conditions. Different relationships may appear among the same constructs under different conditions. This would not necessarily be seen as contradictions or refutation, but would instead add richness to descriptions of the phenomena.

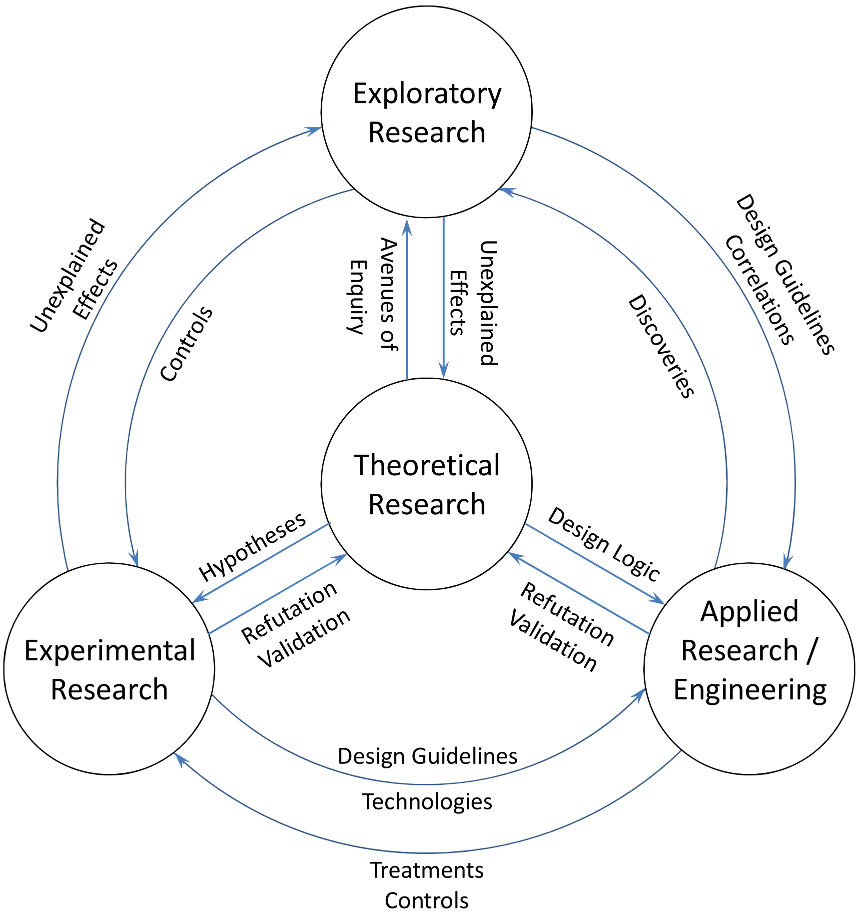


Fig. 1. Each Mode of Scientific Inquiry Informs the others. All for modes, therefore, can be brought to bear on the objects of inquiry for DSR to improve the richness and rigor of DSR findings. (Arrows are illustrative examples. Full articulation could occupy many pages.).

Standards of Rigor for Exploratory Research: The validity and generalizability of exploratory findings are established through *concatenation* - the accumulation of studies from which inductions may be made [13, 17], and by which inter-subjective concurrence on inductions may be established [18]. DSR may concatenate, for example, by testing related solutions to a class of problems across multiple domains. Definitions of constructs should be sufficiently explicit to demarcate them from other closely related constructs.

Exploratory research do not provide logic by which causality may be established. If two constructs correlate, it could be that the first causes the second, the second causes the first, or that some third unknown construct causes both [15]. The only logic for distinguishing among these possibilities is in theoretical and experimental research.

In exploratory papers, therefore, statements of relationships among constructs should be expressed in the language of association, e.g., *A is strongly associated with B; C correlates with D;* or, *E is inversely related to F.* Statements in exploratory models should exclude language that connotes causation, avoiding terms like *influences, impacts, affects*, *determines,* or *causes*. When discussing their models, however, exploratory researchers may, propose carefully qualified conjectures about *possible* causal relationships among the phenomena they describe, e.g. *G may influence H; I may be a function of J.*

Criteria for Exploratory Research Contributions: To be a contribution, an exploratory study should a) describe newly discovered phenomena and/or unreported contexts under which phenomena vary; or b) should concatenate previous findings , up to the point of conceptual saturation, where further exploratory studies yield no new insights [13].

Contributions of Exploratory Research to Other Modes of Inquiry: Exploratory research provides the foundation for all other modes of inquiry. It discovers the phenomena that theoretical research should explain. It's discoveries also inform experimental researchers about effects for which experimental designs should control. Its findings yield insights to AS/E researchers about the people, the problems and opportunities, and the environments that drive AS/E. Its correlative models let AS/E researchers predict possible consequences of design choices. Case studies of design projects yield design guidelines and best practices that inform design theories [19] (Figure 1).

DSR Cycle Activities as Exploratory Research: DSR Relevance, Design, and Rigor Cycles activities can be realized as exploratory research. For example, identifying problems, opportunities, stakeholders, goals, design drivers, constraints, and requirements during the design cycle corresponds to the discovery of phenomena and descriptions of the contexts in exploratory research. When solutions *have been derived by intuition,* field-testing in the Relevance Cycle constitutes primary exploratory research. In the Rigor Cycle, informing design choices with exploratory reports of correlation or association is, by definition AS/E research. Validation of such solutions constitutes exploratory concatenation.

3 On DSR Activities as Theoretical Research

The goal of theoretical research is to create models of cause and effect that predict and explain variations in phenomena. The phenomenon-of-interest in theoretical research is always an effect, never a cause. In DSR, the phenomena of most interest would be the outcomes the designer seeks to improve with designed artifacts. DS researchers need kernel theories that can predict and explain the effects of contemplated design choices.

The Products of Theoretical Research: The product of theoretical research is a *deductive nomological theory* that predicts and explains variations in a phenomenon. These are sometimes called causal, formal, or explanatory theories; Gregor [20] calls them "theories that predict and explain;" Stebbins [13] calls them received theories.

The term, "theory," however, is overloaded, being also attached to other kinds of models besides deductive nomological theories – taxonomies, descriptive models, grounded theories, and design theories among them [20]. Each kind of theory models a different aspect of reality. Each has different kinds of statements, represents different relationships, has different standards of rigor, and serves a different purpose. These kinds of theory are useful to science, but are not the product of theoretical research. Descriptive and grounded theories are, as noted, the product of exploratory research. Design theories are a product of AS/E.

A deductive nomological theory has two kinds of statements (sometimes called covering laws or general laws): axioms[1] and propositions. A theoretical *axiom* states an assumption about mechanisms that could give rise to a phenomenon. For example, to explain user productivity, one might begin with an assumption like:

> ***Axiom 1****. Human attention resources are limited.*

A *theoretical proposition* is a functional statement of cause-and-effect between two constructs. A construct is an abstract concept that represents a causal or consequent element in the environment. It should be possible to a derive theoretical propositions from its axioms by internally consistent deductive logic. For example:

> *If, as Axiom 1 posits, human attention resources are limited, and if productive effort requires attention, then it would have to be that:*

> ***Proposition 1:*** *User productivity is an inverse function of distraction.*

In DSR, the logic of a nomological theory can be used to predict and explain the effects of design choices. If Proposition 1 holds, for example, then a DS researcher should be able to improve user productivity by eliminating distractions from the system, and/or by using the system to mitigate distraction in the environment.

Standards of Rigor for Theoretical Research: Proposition should express causal relationships between constructs. Axioms should propose mechanisms that could account for the phenomena-of-interest. It should be possible to derive its propositions from its axioms by deductive logic. It should be possible to falsify the constructs and propositions of a theory by experience [18, 21]. Propositions should not be tautological (true by definition, or by circular reasoning). Definitions of causal constructs should be sufficiently explicit that one could devise treatments that instantiate differing values of the causal construct [21]. Definitions of consequent constructs should be sufficiently explicit that they can be measured in an operationally specific manner [21]. The term, satisfaction, for example, has been attached to both judgments and emotions in the IS literature. Definitions of satisfaction would therefore have to clarify that distinction. The construct, *outcomes*, which has appeared in many IS theories, is not sufficiently specific because it could refer to every phenomenon in the IS domain.

---

[1] The term, *axiom*, has other connotations in other contexts. Some *authors* apply the term to theoretical positions that have accreted massive and unequivocal empirical support (e.g., F=MA). Others use the term to mean, "that which is widely assumed to be true."

The generalizability of a theory is the range of contexts to which it can be applied. A more-specific theory may explain a phenomenon in a given context or under a bounded range of conditions, and may do so in terms more closely related to the context, making it easier to apply the theory in that context. At the same time, specificity limits the theory's generalizability. For example, early IS Satisfaction theories that included attributes of specific technologies were useful for predicting satisfaction with those objects, but did not generalize well to new technologies. More-general disconfirmation theories of satisfaction [22] explained satisfaction with any technology at the time outcomes were realized, but could not account for effects long before or after outcomes were obtained. Yield Shift Theory [23] is still more general, explaining satisfaction with any objects in any contexts (although it has not yet accrued sufficient empirical support to establish its scientific utility). At the same time, it may require more reasoning to apply a general theory to a specific case.

Note that it is neither required, nor logically possible to derive or defend the axioms of a theory. They are assumptions, and are deemed to be received [18]; their origins are not relevant to the logic of the theory.

Criteria for Theoretical Research Contributions: A nomological theory contributes to knowledge if its scientific utility or parsimony are greater than those that preceded it [21]. A theory has more utility if it accounts for more variations in a phenomenon in more contexts; having more explanatory power, it is a contribution to theoretical research. The parsimony of a theory is the number of constructs and statements it requires to achieve its explanatory power [18]. A new theory with same explanatory power, but fewer constructs or statements would be a contribution to theoretical research. If, however, adding more constructs, axioms, or propositions to a theory were to *increase* its explanatory power, then it would be deemed a contribution, even if it were less parsimonious.

Contributions of Theoretical Research to Other Modes of Inquiry: Theoretical research often anticipates effects not yet observed, suggesting fruitful lines of inquiry to exploratory research. Theoretical research is the *raison d'être* for experimental inquiry, which has as its purpose to falsify theoretical propositions [18]. To AS/E, theoretical research sends explanations with which designers can predict the consequences of new design choices (Figure 1). Theories may thus become design guidelines; e.g. if, as YST proposes, satisfaction is a function of shifts in yield for the active goal set, then UI/UX designers could invoke satisfaction responses with design choices that impact the perceived likelihood and/or utility of goal attainment [23].

DSR Cycle Activities as Theoretical Research: In the DSR Rigor Cycle, design choices may be informed by a nomological deductive kernel theory [5]. If existing theory does not explain the outcomes of interest, the DS researcher may improve existing theory or derive a new theory for that purpose, e.g., [23]. Doing so contributes to theoretical research. In the Design Cycle, validating an artifact derived from a theory would test that theory.

4 On DSR Activities as Experimental Research

The goal of Experimental research is to test the propositions of a deductive nomological theory. It may also be called *confirmatory research* [13], but confirmation should not be misinterpreted as proof; scientific method provides no logic by which a theory may be proven true. Results that are consistent with a theoretical proposition may support a proposition, but do not prove it. By the same token, no single experiment can claim to have broken a proposition. There are many threats to the validity of an experiment [24], and it is not possible to control for all of them in a single study. It therefore requires a body of experimental work by a community of researchers to credibly support or refute a theory.

Products of Experimental Research: The products of experimental inquiry are hypotheses, experimental designs and methods, and analyzed data sets. The term, hypothesis has several connotations in the scientific literature; it is sometimes used as a synonym for the terms, prediction, conjecture, and proposition [25]. In experimental research, a *hypothesis* is a comparative statement that contrasts the value of a dependent variable across treatments that instantiate differing values of an independent variable. A dependent variable always instantiates the consequent construct of a theoretical proposition. In DSR it is a measure of an outcome the designer seeks to improve with the artifact, and so measures the degree to which design objectives have been achieved. An independent variable always instantiates the causal construct of a proposition. In DSR, one of the treatments is likely to be a theoretically-informed designed artifact. Another treatment may be a previously designed artifact, or a control condition where no technological artifact is introduced. Hypotheses should be derived by internally consistent deductive logic from the theoretical propositions they are meant to test. For example, to test the Distraction proposition above, one could reason as follows: *If, as Proposition 1 states, end-user productivity is an inverse function of distraction, then it would have to be that:*

> ***Hypothesis 1.*** *People using a digital brainstorming tool that plays video clips of exuberant dancers at random intervals will produce fewer useful ideas than will people who use a tool that plays no clips of dancers.*

In H1, the clip treatment is a high value for distraction; the lack-of-clips a low value.

Standards of Rigor for Experimental Research: Many issues of validity surround Experimental research. This section only lists a small but important subset: construct validity, internal validity, external validity, and experimenter bias [24]. Construct validity is the question of whether the variables used in the hypotheses actually instantiate the constructs in the proposition. Science has no definitive proof for construct validity, but statistical tests for convergent and discriminant validity are at least useful for excluding some flawed measures [24]. Internal validity is the question of whether the observed results were actually caused by the experimental treatment instead of by something else. Numerous disciplines pertaining to experimental designs and controls should be brought to bear to improve internal validity (see [24]). External validity is the degree to which results of the experiment would generalize to contexts other than those of the experimental conditions. If for example, a DS experiment user interface color were run with two-color monitors, the study would have low external validity since results might differ on commonly-used monitors that display 16 million colors [24].

Experimenter bias is the question of whether the experimenter's expectations, preferences, actions, omissions, or limitations skewed experimental results. There is a widespread misconception that the philosophy of science considers the scientist to be objective. On the contrary, causal epistemology assumes the observer is subjective [18]. The validity of any finding is therefore in question until an experiment has been replicated by other subjective observers under other conditions. Inter-subjective concurrence – all subjective observers obtaining similar results – provides some assurance that outcomes may be sound [18].

Studies that *measure* the independent variable, rather than manipulating it with treatments, do not conform to the logic of experimental research, so no causation may be inferred from the results. Such studies are exploratory; to minimize confusion they should be labeled as *investigations* or *explorations* rather than as experiments.

Criteria for Experimental Research Contributions: Experiments contribute to scientific knowledge if a) hypotheses were derived from theoretical propositions by sound deductive logic; b) construct validity is reasonably argued; c) experimental design rules out most alternative explanations for the results, and threats to validity that could not be controlled are noted; d) Statistical analyses are a sound test of the hypotheses, e) the analyses support the hypotheses, and f) the literature is not already saturated with replicated studies supporting the proposition being tested. Negative experimental results may also contribute to science if a) the first four conditions above hold; b) statistical analysis reveal very high statistical power (had there been an effect, the study would have been likely to reveal it); and c) the literature contains robust empirical support for the proposition. This would be a credible challenge to a generally accepted position, and so worthy of further attention from the scientific community.

Contributions of Experimental Research to Other Modes of Inquiry: Experiments sometimes reveal previously unknown phenomena and patterns of correlation, and contribute to exploratory research. Negative experimental findings sometimes inform ways to improve a theory, and so contribute to theoretical research. Positive findings build support for a theory, increasing its value to society, and so contribute to theoretical research. When experimental findings inform design processes and choices, or validate artifacts, they contribute insights to AS/E.

Although the only purpose of Experimental *research* is to test formal theoretical propositions, experimental *techniques* are also useful in Exploratory and AS/E. An exploratory study based on experimental techniques can reveal new phenomena and new details about known phenomena, even though its results cannot be interpreted as having tested a theoretical proposition. Likewise, the findings of an experimental validation of a DSR artifact inspired by intuition would be both a contribution to AS/E, in that they validate the new solution, and a contribution to exploratory research, in that they explore phenomena in previously unexamined contexts and conditions. Such findings, though they would be regarded as AS/E and exploratory respectively, they would not be contributions to experimental research because they do not test a theoretical proposition. When experimental techniques are used in exploratory or AS/E studies, hypotheses should not be advanced, because there will be no theoretical propositions from which to derive them. One can use instead research questions or conjectures. A research question would convert hypothetical language to a question (e.g., RQ1. Will people who use an electronic…score higher on…than people who use…?). A conjecture would differ linguistically from a hypothesis in label only, (e.g., *Conjecture 1. People who use an electronic…will score higher on… than people who use…*). The conjecture label will show readers that the author knows the study does not test a theoretical proposition, and so may preclude them from demanding experimental rigor for a study where it would not be logically or philosophically warranted (Figure 1).

DSR Cycle Activities as Experimental Research: In the Rigor Cycle, when one draws on a theory to inform design choices, that frames a treatment for an experimental hypothesis. In the Relevance Cycle, validating a theoretically-informed artifact by comparing it to a prior solution could be an instance of an experiment on a hypothesis derived from the theory. Positive findings would both validate the artifact, and support the theory.

5 On DSR Cycles as Applied Research/Engineering

The goal of AS/E research is to use scientific knowledge to solve important practical problems. AS/E is distinguished from engineering practice in that engineering practice seeks to create a specific instance of a useful artifact to solve a specific problem, (e.g., [26, p. 86]), while AS/E seeks to create novel, generalizable solutions for an important class of problems, and to synthesize bodies of knowledge, construction principles, and generalizable work practices that can increase the likelihood that designed artifacts will meet design objectives. In DSR, the synthesized knowledge would include the kernel theories and other findings that could inform the design choices. Construction principles encompass structure and function of existing and possible technology [26, p. 90]. The generalizable work practices would be engineering methodologies. These contributions can, over time, be codified into a *design theory* (DT) that a) defines the purpose and scope of a design methodology; b) identifies principles of form and function for design solutions in that scope; c) defines criteria for generalizability of solutions by identifying requisite variety that a solution in the scope should accommodate (called “artifact mutability”); d) identifies justificatory knowledge in the form of kernel theories and other knowledge that can inform designs in the scope; e) provides guidelines for implementation; and f) provides an expository instance of a solution in the scope [19].

Where theoretical inquiry seeks relationships in the form, *A causes B*, the logic of AS/E is, *If you want to achieve B, then you should do A*. German technology philosopher Kornwachs [27, p. 72] condenses Bunge’s [28] pragmatic syllogism more concisely as: *If A --> B*. This expression means: *Under certain circumstances, realizing State A will cause State B to exist. If State B is desired, then try to bring about State A*. While *A causes B* should be true in all contexts, *If you want to achieve B, then do A* is dependent on its specific socio-technical context for two reasons. Firstly, there are many interacting conditions other than the designed artifact that may affect its utility for achieving *A*. An artifact requiring electrical power, for example, may be deployed in an environment without electricity, and so be incapable of creating *A*. Likewise, a good artifact may be used badly, and so not produce *A*, despite its potential. In Engineering research, therefore, is often useful and necessary to inform design choices by a mixture of scientific knowledge (from the natural and social sciences), intuition, empirical knowledge (e.g., from tests) and prior technical knowledge [29].

Products of AS/E Research: The research products of AS/E research are: a) detailed descriptions of important classes of problems, and the contexts in which they emerge; b) generalizable design objectives, constraints, and requirements for addressing a class of problems; c) generalizable solutions for a class of problems, e.g., design patterns [30, 31]; d) expository instances of generalizable solutions; e.g., reference models [32], and proof-of-concept prototypes [1]; f) evidence that solutions are useful and generalizable; and g) the elements that comprise design theories for implementing solutions for a class of problems, e.g., methods such as object-oriented analysis and design [33].

Standards of Rigor for AS/E Research: Where an exploratory researcher says, "Gee, that's *funny*," (Isaac Asimov quoted by [34]) and a theoretical researcher shouts, "Eureka!" the successful AS/E researcher exclaims. "It works!" [26. p. 97]. The principle criterion for a contribution to AS/E knowledge is its usefulness. As with theoretical contributions, AS/E contributions should be original, generalizable, and validated. Originality may be established by comparing contributions to the state of the art. Generalizability may be established by demonstrating the applicability of the solution to a range of contexts. Validity may be justified by the evaluation of the results [35]. Justification efforts could include pilot tests in the natural environment [36], experiments, expert evaluations, or, in some cases, the consensus of the scientific community.

Criteria for AS/E Research Contributions: AS/E Research applies scientific knowledge to solve important practical problems. It is by definition, therefore, informed by the other three modes. AS/E, however, also informs the other three. An AS/E researcher who investigates a previously unexamined domain *to* identify its problems, opportunities, constraints could make a contribution to exploratory research. Likewise, an AS/E researcher who tests a new design inspired by intuition rather than theory may be conducting exploratory research. When an AS/E researcher develops or improves a theory to better inform design choices, the resulting model could be a contribution to theoretical research. An AS/E researcher who validates a theory-informed artifact with an experiment, contributes to experimental research.

Contributions of AS/E Research to Other Modes of Inquiry: When an applied researcher *discovers* a previously unreported effect, that may be a contribution to exploratory research. If the applied researcher develops a theory to explain an effect in order to inform design choices, that would be a contribution to experimental research. When an applied research validates a technology whose design choices were informed by a theoretical proposition by comparing it to an earlier solution, the results may be a contribution to experimental research. If the experiment fails to support the theory, that may also be a contribution to exploratory research.

DSR Cycle Activities A AS/E: Because the roots of DSR are in engineering, the activities of the DSR closely parallel those of AS/E; indeed, the current framing of DSR may be seen as a domain-specific reinvention of AS/E. The parallels can be demonstrated by considering three key activities of AS/E:

1. Identify an important class of unsolved problems. This corresponds to the *Requirements* activity of the DSR Relevance cycle, where the current state and desired state are identified by identifying the actors, their goals, the design objectives and key design drivers and constraints. AS/E outputs, like DSR requirements, should be generalizable to a class of problems and a range of contexts .
2. Design generalizable solutions. This AS/E activity corresponds closely with the *Design and Build Artifacts* activity in the DSR Design Cycle, where theories, intuition, and prior engineering knowledge are used to produce classical *engineering* outputs: models, methods, and expository instances of a generalizable solution.
3. Validate the solution. This AS/E activity maps directly to the *Validate* and *Field Test Artifacts* activity of the DSR Relevance cycle. As in AS/E, DSR validation consist of an empirical test of the designed artifact or process with, for example, experimental techniques. Where exploratory or theoretical knowledge is realized in the artifact, the validation produces research spillovers in those domains (Figure 1).

6 On Expanding the Scope of Design Science Inquiry

6.1 Increasing the Rigor of DSR by Expanding Its Scope

The preceding sections define the goals, research products, and standards of rigor for each mode of exploratory, theoretical, experimental, and AS/E inquiry, and demonstrate that DSR Cycle Activities can be realized as instances of one or more of the four modes of scientific inquiry. Depending on the needs of the DS researcher and the phase of the research, Relevance Cycle activities like *Identify Requirements* and *Do Field Tests,* can be realized variously as exploratory, theoretical, experimental, and AS/E modes of inquiry. Various aspects of the *Build Design Artifacts and Processes* activity in the Design Cycle can be realized across as instances of all four modes. The Rigor Cycle activity, *Ground design in applicable knowledge,* corresponds exactly to the AS/E activity, *Identify relevant scientific knowledge.* Elements of the Rigor Cycle's *Add Knowledge to Knowledge Base* activity can be realized in each of the four modes of scientific inquiry, because all activities of the four modes of scientific inquiry contribute to the knowledge base, and all are instantiated by one or more DSR Cycle activities.

Given that all activities of the DSR Cycles Model can be realized as one or more instances of the four modes of scientific inquiry, and given that all four modes have accepted standards of rigor, and assuming that the DS researcher implements an activity under the disciplines of a mode, the standards of rigor for the four modes of inquiry can be used to defend the rigor of DSR. Expanding the scope of DSR to bring all four modes of scientific inquiry to bear on the DSR objects of inquiry would therefore increase the rigor of DSR. It is important to note, however, that all DSR activities can also be realized in ways that do not conform to the logic and disciplines of any of the four modes of scientific inquiry. In such cases, the rigor of the DSR could not be defended by the standards of rigor for the modes. Given that lack of rigor, however, one might argue that such activities are not, in fact, design *science*.

6.2 Increasing the Richness of DSR by Expanding Its Scope

The preceding sections also demonstrate that there is a synergy among the four modes of scientific method because each mode informs the others; none stands on its own (Figure 2). Increasing the scope of DSR beyond its engineering roots so as to bring all four modes to bear on the DSR objects of inquiry would therefore increase the depth and variety of DSR's normal AS/E contributions, increasing the strength of DSR's foundation. It would also initiate a fresh stream of DSR contributions to exploratory, theoretical, and experimental research.

7 On Cycles Model for DSR Informed Four Modes of Scientific Inquiry

If DSR activities are realized as instances of the four modes of scientific inquiry, then the *activities* and their findings will be scientifically rigorous. Under those conditions there is no need for a separate Rigor Cycle in a DSR cycle model. If realized rigorously, all DSR activities have the potential to contribute to the knowledge base. Given that stakeholders could be involved in all DSR activities, all activities could require interactions with the environment. We therefore propose a modified DSR Cycles Model for activities based on the four modes inquiry (Figure 2). It characterizes DSR as three activities: 1) Discover Problems and Opportunities; 2) Design and Build Artifacts and Processes; and 3) Validate Artifacts and Processes.

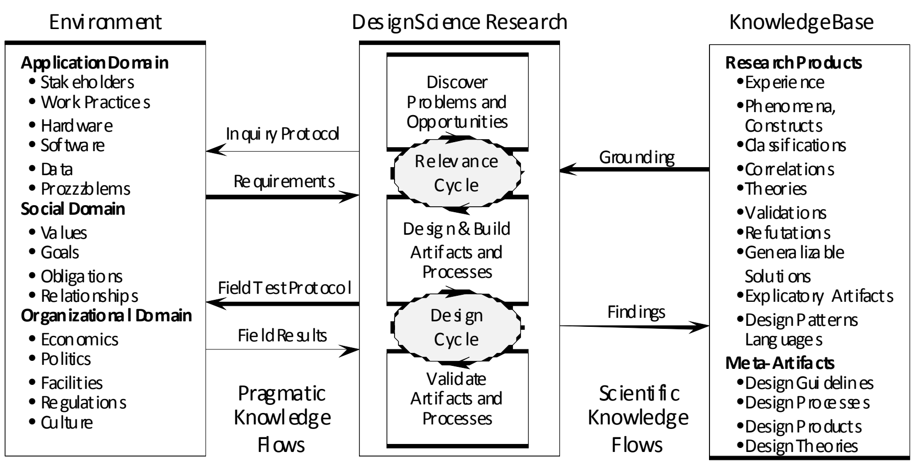


Fig. 2. A MODIFIED DSR CYCLES MODEL for Activities Informed by the Four Modes of Scientific Method: Exploratory, Theoretical, Experimental, and Applied Science/Engineering. If all DSR activities are conducted with scientific rigor, there is no need for a separate Rigor Cycle. Arrows signify information flows among DSR activities, the environment, and the knowledge base. (Lists of concepts are exemplary rather than exhaustive.).

The model depicts a Relevance Cycle between the Discovery and Design activities, and a Design Cycle between the Design and Validate Activities. It signifies that any DSR activity may draw from or add to the Knowledge Base. It further signifies that any DSR activity may engage with stakeholders in the environment to learn more about their problems and opportunities and to involve them in the DSR process. This model reflects the rigor and richness that can be gained by broadening the scope of DSR modes of inquiry. The arrows signify flows of knowledge between DSR activities, the knowledge base, and the environment.

8 Conclusions

This paper argued that DSR activities can be realized as instances of four modes of scientific inquiry: exploratory, theoretical, experimental, and AS/E. It shows a synergy among the four modes of inquiry, because each mode of inquiry informs the other three. To exclude any of them from DSR, therefore, is to impoverish that body of research. It is consequently important to the advancement of DSR to expand the scope of DSR beyond its engineering roots to embrace all four modes of scientific inquiry.

We argue that, because DSR *activities* can be realized as instances of the four modes of inquiry, it is possible to defend the rigor of DSR activities in terms of the goals, research products, standards of rigor, and criteria for contributions to knowledge that have already been established and accepted for these modes of inquiry. The paper demonstrates this position by enumerating aspects of each mode of inquiry and linking them to DSR activities. It would be useful to the advancement of DSR, therefore, to execute DSR activities according to the precepts and disciplines of the established modes of inquiry until such time as other means of defending its rigor may be established.

References

1. Hevner, A., March, S., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quart. 28(1), 75–105 (2004)
2. Venkatesh, V., Brown, S., Maruping, L., Bala, H.: Predicting Different Conceptualizations of System Use: The Competing Roles of Behavioral Intention, Facilitating Conditions, and Behavioral Expectation. MIS Quart. 32(3), 483–502 (2008)
3. Titah, R.: Nonlinearities Between Attitude and Subjective Norms in Information Technology Acceptance: A Negative Synergy? MIS Quart. 33(4), 827–844 (2009)
4. Karr-Wisniewski, P., Lu, Y.: When More is Too Much: Operationalizing Technology Overload and Exploring its Impact on Knowledge Worker Productivity. Comput. Hum. Behav. 26(5), 1061–1072 (2010)
5. Hevner, A., Chatterjee, S.: Design Science Research: Looking to the Future. In: Hevner, A., Chatterjee, S. (eds.) Design Research in Information Systems, pp. 261–268. Springer, New York (2010)
6. Iivari, J.: A Paradigmatic Analysis of Information Systems as a Design Science. Scand. J. Inform. Syst. 19(2), 39–64 (2007)

7. Schwaber, K., Beedle, M.: Agile Software Development with Scrum, vol. 18. Prentice Hall, Upper Saddle River (2002)
8. Booch, G., Rumbaugh, J., Jacobson, I.: The Unified Modeling Language User Guide. Addison-Wesley Object Technology Series. Addison-Wesley Professional, Reading (2005)
9. Chen, P.: The Entity-Relationship Model—Toward a Unified View of Data. ACM Transactions on Database Systems (TODS) 1(1), 9–36 (1976)
10. Codd, E.: A Relational Model of Data for Large Shared Data Banks. Commun. ACM 13(6), 377–387 (1970)
11. Peffers, K., Tuunanen, T., Rothenberger, M., Chatterjee, S.: A Design Science Research Methodology for Information Systems Research. J. Manag. Inform. Syst. 24(3), 45–77 (2007)
12. Indulska, M., Recker, J.: Design Science in IS Research: A Literature Analysis. In: Gregor, S., Ho, S. (eds.) Proceedings of the 4th Biennial ANU Workshop on Information Systems Foundations, Canberra, Australia, October 2-3, pp. 1–13 (2008)
13. Stebbins, R.: Exploratory Research in the Social Sciences. Sage Pubs, Inc., Thousand Oaks (2001)
14. Vogt, P.W.: Dictionary of Statistics & Methodology: A Nontechnical Guide for the Social Sciences, 2nd edn. Sage, Thousand Oaks (1999)
15. Sharpe, P.B.: A Critical Analysis of the Canons of Science. Philos. Sci. 7(2), 159–167 (1940)
16. Adam, M.: Variables Affecting Information Technology End-User Satisfaction: A Meta-Analysis of the Empirical Literature. International Journal of Human-Computer Studies 52(4), 751–771 (2000)
17. Lofland, J.: Doing Social Life: The Qualitative Study of Human Interaction in Natural Settings. Wiley, New York (1976)
18. Popper, K.: The Logic of Scientific Discovery. Psychology Press, San Diego (2002)
19. Gregor, S., Jones, D.: The Anatomy of a Design Theory. J. Assoc. Inf. Syst. 8(5), 312–335 (2007)
20. Gregor, S.: The Nature of Theory in Information Systems. Manag. Inform. Syst. Quart. 30(3), 611–642 (2006)
21. Bacharach, S.: Organizational Theories: Some Criteria for Evaluation. Acad. Manage. Rev. 14(4), 496–515 (1989)
22. Oliver, R.: A Cognitive Model of the Antecedents and Consequences of Satisfaction Decisions. J. Market. Res. 17(4), 460–469 (1980)
23. Briggs, R., Reinig, B., de Vreede, G.: The Yield Shift Theory of Satisfaction and its Application to the IS/IT Domain. J. Assoc. Inf. Syst. 9(5), 267–293 (2008)
24. Cook, T., Shadish, W., Campbell, D.: Experimental and Quasi-Experimental Designs for Generalized Causal Inference. Houghton Mifflin, Boston (2002)
25. Godfrey-Smith, P.: Theory and Reality: An Introduction to the Philosophy of Science. University of Chicago Press, Chicago (2003)
26. Konig, W.: Goals of Technological Sciences (In German: Ziele der Technikwissenschaften). In: Banse, G., Grunwald, A., Konig, W., Ropohl, G. (eds.) Cognition and Design - A Theory of the Technological Sciences (in German: Erkennen und Gestalten - eine Theorie der Technikwissenschaften), pp. 84–98. Edition sigma, Berlin (2006)
27. Kornwachs, K.: Technological Knowledge (In German: Technisches Wissen). In: Banse, G., Grunwald, A., Konig, W., Ropohl, G. (eds.) Cognition and Design - A Theory of the Technological Sciences (in German: Erkennen und Gestalten - eine Theorie der Technikwissenschaften), pp. 71–84. Edition sigma, Berlin (2006a)
28. Bunge, M.: Scientific Research. Searching for Truth, vol. I. Springer, Berlin (1967)

29. Jobs, E.: General Problems of Technological Cognition (in German: Allgemeine Probleme technikwissenschaftlicher Erkenntnis). In: Banse, G., Grunwald, A., Konig, W., Ropohl, G. (eds.) Cognition and Design - A Theory of the Technological Sciences (in German: Erkennen und Gestalten - eine Theorie der Technikwissenschaften), pp. 184–239. Edition sigma, Berlin (2006)
30. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object. Addison-Wesley, Reading (1995)
31. Alexander, C.: The Timeless Way of Building. Oxford University Press, New York (1979)
32. Scheer, A.W.: Business Process Engineering - Reference Models for Industrial Enterprises. Springer, Heidelberg (1994)
33. Booch, G., Maksimchuk, R., Engle, M., Young, B., Conallen, J., Houston, K.: Object-Oriented Analysis and Design With Applications. Addison-Wesley, Reading (2007)
34. Meyer, C.: Seeking “Asimov Moments.” Now, New Next Blog. Harvared Business Review (2008), http://blogs.hbr.org/now-new-next/2008/10/welcome-to-now-new-next.html (accessed January 28, 2011)
35. Frank, U.: Die Konstruktion möglicher Welten als Chance und Herausforderung der Wirtschaftsinformatik. In: Becker, J., Krcmar, H., Niehaves, B. (Hrsg.) Wissenschaftstheorie und gestaltungsorientierte Wirtschaftsinformatik, pp. 167–180. Physica, Heidelberg (2009)
36. Schwabe, G., Krcmar, H.: Piloting a Sociotechnical Innovation. In: Proceedings of the 8th European Conference on Information Systems ECIS 2000, pp. 132–139 (2000)

A Design Science Research Roadmap

Ahmad Alturki, Guy G. Gable, and Wasana Bandara

Queensland University of Technology, Information Systems Discipline, 126 Margaret St., Brisbane, Queensland, Australia

Abstract. This paper proposes and synthesizes from previous design science (DS) methodological literature a structured and detailed DS Roadmap for the conduct of DS research. The Roadmap is a general guide for researchers to carry out DS research by suggesting reasonably detailed activities. Though highly tentative, it is believed the Roadmap usefully inter-relates many otherwise seemingly disparate, overlapping or conflicting concepts. It is hoped the DS Roadmap will aid in the planning, execution and communication of DS research, while also attracting constructive criticism, improvements and extensions. A key distinction of the Roadmap from other DS research methods is its breadth of coverage of DS research aspects and activities; its detail and scope. We demonstrate and evaluate the Roadmap by presenting two case studies in terms of the DS Roadmap.

Keywords: Design Science, Design Science research Roadmap, Design Science research methodology, Design Research, Information System Design Theory, Archival Analysis.

1 Introduction

Design science (DS) research has become an accepted approach for research in the IS discipline [1, 2], with dramatic growth in recent, related literature[1] [6]. Though this literature reflects healthy discussion around the balance of rigor and relevance in DS research, consensus on even the fundamentals (e.g. DS Definition, DS methods, DS Outputs) has yet to be achieved [7]; the area still being in its genesis [2, 8].

Views and prescriptions on the methodology of DS research appear particularly disparate, e.g. [4, 9-17]. One set of guidelines by Hevner et al. [18] has been widely cited, there being concern however with their high-level and lack of specificity. Archival analysis by Indulska and Recker [19] of papers reporting studies that purportedly conform to the Hevner et al. [18] guidelines, reveals few instances of their actual application. Similarly, and more recently, Venable [20] investigates the opinions of IS scholars on the importance of the Hevner et al. [18] guidelines, observing “extensive disagreement on what guideline areas should be used as criteria and standards for evaluation [of DS research]” [20]. Walls et al. in [21] noticed also that few papers

[1] Strong, relatively recent interest in DS [2, 3] has stimulated journal special issues (e.g. 2008 MIS Quarterly vol. 32 no. 4 [4]); specialized conferences in the area (e.g. DESRIST begins in 2006); and the publication in 2007 of the first textbook on ISDR methods [5].

H. Jain, A.P. Sinha, and P. Vitharana (Eds.): DESRIST 2011, LNCS 6629, pp. 107–123, 2011.

use their Information System Design Theory (ISTD) proposed in [22] despite this paper discussing how to develop theory being frequently cited.

Thus, though generally highly regarded and widely cited, DS methodological guidance from e.g. Hevner et al. [18] and Walls et al. [22] is seldom 'applied', suggesting that existing guidelines and methods are insufficiently clear, or inadequately operationalised - still too high a level of abstraction [14]. This lack of detailed, more specific guidance, becomes stark in comparison with research methodology in the behavioral sciences, where guidance on methods e.g. experimentation (e.g. [23]) and analytical techniques e.g. structural equation modeling (e.g. [24]), has evolved to become highly prescriptive and specific. And though the dangers of overly constraining designers' imaginations and creativity through prescription is recognized (e.g. [1]), there would too seem to be a general consensus that some level of generality of approach and more detailed guidance is possible and in some areas of DS research more extensively possible; in other words, that we have not gone far enough. Winter [4] states there is a "lack of a commonly accepted reference process model for design research", suggesting that a more complete methodology is a key lack in DS research [7]. This paper is motivated by this lack and the authors believe that there is need for a structured DS Roadmap to guide researchers across the DS lifecycle.

This paper proposes and synthesizes from existing DS methodological writings a structured and detailed DS Roadmap for the conduct of DS research. The DS Roadmap is a guide, providing detailed steps for researchers to perform DS research. Though highly tentative, it is believed the Roadmap usefully inter-relates many otherwise seemingly disparate, overlapping or conflicting concepts. It is hoped the DS Roadmap will aid in the planning, execution and communication of DS research, while also attracting constructive criticism, improvements and extensions. A key distinction of the Roadmap from other DS research methods is its breadth of coverage of published DS research concepts and activities; its detail and scope. We next describe the process employed for archival analysis of past DS research, subsequently presenting the synthesized Roadmap illustrating its main components and steps. Thereafter, two case studies are used to show the value of the DS Roadmap, followed by conclusions.

2 Methodology

The main source of concepts for the intended Roadmap was existing DS literature. Searching employed a snowball approach, starting with highly cited papers - e.g. [13, 18], gradually fanning out to other relevant publications, and paying particular attention to related special issues and specialist conferences. Closer attention was paid to papers having a considerable citation rate, taking into account the year of publication [3]. These include articles that are largely methodological, as well as articles that are methodological in part only. Through this process, 60 key articles were compiled from which the Roadmap concepts are drawn (a list of the full set of these papers is available from the 1st author).

The 60 articles were loaded into NVivo 8.0 software for content analysis. As key concepts were identified, they were entered as nodes in separate classification and glossary trees. A glossary of DS-related concepts and definitions was compiled.

Having thoroughly read and partially codified the 60 articles, and having populated the glossary of concepts and terms, a further pass on the dataset revealed four main themes:

DS as an Approach. 'how to', guidelines for conducting, or output of DS e.g. [9, 14].

DS Philosophy. and how it differs from other methods, e.g. [16][16][8, 16].

The Role of Theory and Theorizing in DS. e.g. [6, 25].

Applications[2] of DS. e.g. [26, 27].

With the goal of synthesizing from the literature a pragmatic and detailed Roadmap to guide DS research, a key node in the Nvivo classification tree was 'DS methods/steps', against which was mapped all material that either explicitly or implicitly suggested guidance on how to undertake DS research. The next section shows the result of synthesizing this node content.

3 The Synthesized Design Science Roadmap

In this section, the relevant literature is discussed chronologically (see Table 1), thereby minimizing repetition and best reflecting the evolution of DS thinking, with key methodological guidance drawing on, influenced by, or referring to prior published ideas[3] – e.g. Hevner et al.'s [18] work draws extensively on March and Smith [13]; Gregor and Jones's [25] work is based in Walls, Widmeyer and El Sawy [22]; and Venable [17] makes much reference to Nunamaker, Chen and Purdin [29]. Certain of these source writings address some specific aspect of DS research activity – e.g. 'evaluation', as in [30]. Others are broader, more abstract and less prescriptive – e.g. [14].

Fig. 1 depicts the Roadmap, as synthesized from the archival analysis, inter-relating methodological contributions across the DS lifecycle, from the early 'spark' of a design idea, through to final publication. The Roadmap is consistence with Hevner et al.'s [18] IS research framework and the three DS research cycles (relevance, design and rigor) as presented in [10]. The authors believe the Roadmap extends Hevner et al.'s framework, by providing detailed steps for conducting DS research, and by indicating how these activities interrelate with the knowledge-base and environment. It thus adds components (B, C and D), as explained in following sections. It incorporates a framework for evaluating risk as proposed in [31], and represents a multi-grounded design research process

---

[2] Though many of the 60 articles refer to applications of DS research, all have some methodological emphasis.

[3] Though broader canvassing of the DS literature (i.e. outside IS) would, we expect, encounter greater complexity, we believe the extensive acknowledgement of prior DS-related thinking by IS authors is one reason why we were able to substantially interrelate ideas and concepts with little regard to differing paradigms and understandings. Further, Gregor and Hevner [28] note that though "there exist differing 'camps' in the IS DSR community," the 'design-theory camp and the pragmatic-design camp, they believe these views are complementary rather that conflicting or competing. Regardless, closer attention to implications of differing paradigms, and expansion of the literature review beyond IS, are both valuable extensions of this study.

and design theory as discussed in [6, 32]. Moreover, it adopts the Information System Design Theory (ISDT) structure proposed in [25], and includes a Central Design Repository (CDR) to document all aspects of the DS research journey. The Roadmap is usefully viewed top-down, and from its center to the left (environment) and to the right (the knowledge-base). Structurally, the Roadmap in Fig. 1 consists of four main interrelated components: (A) DS research cycles; (B) DS research output; (C) DS risk management; and (D) Central Design Repository (CDR). Component A gradually feeds and reads from component D which ultimately contributes to component B. Component C and A should execute in parallel, both again using component D. Consequently, components B and D are the sources that contribute to both the environment and the knowledge-base. Following, each of the four main structural components of the Roadmap is discussed in more detail.

3.1 The Design Science Research Cycles

The Roadmap adopts Hevner's [10] three major DS research cycles: rigor, relevance and design (see top of Fig. 1). Discussion in this section focuses on the more detailed 'Design' cycle, pictured vertically at the centre of Fig. 1, design and evaluation constituting the core of DS activity [19].

Table 1 loosely inter-relates DS lifecycle related activities as distilled from 15 relatively more methodological articles. The number of steps proposed in these articles varies with focus, level of detail, and scope of the writings. Some refer to a single step – e.g. evaluation, as in [30]. Others have two rows in Table 1, as they presented their steps in two levels, e.g. [22]. Two papers do not specifically report DS activities [22, 25], but rather propose the structure of design theory as an output of DS research; they are included to assure necessary attention to these outputs and related activities. The centre of Fig. 1 depicts the fourteen main activities of the Design Cycle. Each is addressed briefly following.

Document the Spark of an Idea/Problem. As shown vertically spanning the left- and right-hand sides of Fig. 1, DS research is informed either by practitioners in an environment, where the needs come from; or by researchers based on the knowledge-base, where possible new solutions or extensions are suggested [10]. Accordingly, the issue of the 'spark' could be seen inductively or deductively, respectively. The spark of DS research is sometimes grounded in theory from other research paradigms, such as behavioural science [13, 18, 21, 25, 32]. An idea for DS research might also come from the creative thinking of a designer who can predict an unforeseen need or opportunity based on his/her experience and identified knowledge [1]. Venable calls these 'ideas sparks' and lists examples in [33].

Investigate and Evaluate the Importance of the Problem/Idea. A key characteristic of DS research is that it resolves an important, previously unsolved problem, for a class of businesses or environments, while making a contribution to the knowledge-base [17, 18]. The value of a new solution may perhaps come from solving a known or expected problem, satisfying needs, or innovating something new for the environment. Design researchers must investigate pre-existing knowledge and solutions to insure they do not simply replicate past work of others or undertake what scholars refer to as 'routine design' [18, 33]. Design research must produce new knowledge which comes from "the number of unknowns in the proposed design which when successfully surmounted provide the new information that makes the effort research and assures its value" [16]. Pre-design investigation could involve consideration of the type of problem; Hevner et al. [18] believe the problems that DS research address are 'wicked'[4] in nature. This investigation may involve searching the existing knowledge-base, or collecting primary data through empirical work such as case studies, interviews, experiments or surveys [6, 18, 25, 32, 35]. Research should stop if the problem has already been solved, or if it is found to be unimportant for the targeted environment. Through this process of investigation, the researcher satisfies the relevance condition for DS research in IS [10], while also addressing generalisability [36, 37].

Evaluate the New Solution Feasibility. The importance of an unsolved problem is not enough to warrant DS research. A critical question to ask here is "Is it possible to produce a new solution?" The target objective might be overly ambitious, and therefore not doable within the DS research timeframe and budget. Feasibility is thus a critical early consideration, in order to increase the likelihood of success.

Define Research Scope. The initial research scope and ultimate objective are defined in this step. Since knowledge from DS research is generated through the design process [16, 38], the scope and ultimate objective are revisited frequently for refinement, as the research evolves. New or different objectives may be established with iteration [14, 16]. Though the specific problem definition is not always compulsory, as some designs are motivated by the creativity of designers [1], a researcher should define an expected problem or need, which the intended design aims to overcome or satisfy.

Resolve Whether within the Design Science Paradigm. Once the scope and main objective are defined, the researcher judges whether the research falls under the DS paradigm[5] or not. This assessment may not be straightforward; as DS research can overlap other areas such as routine design and action research e.g. see [6, 8, 9, 11, 13, 18, 22, 29, 33]. Researchers must understand their objective precisely, and compare it to the DS paradigm, on the one hand to insure they intend doing DS research [41], and on the other hand to discover the value of their design.

Establish Type (IS Design Science VS IS Design Research). DS research in IS can be seen as one or both of two types: (1) IS design science and (2) IS design research. Winter [7] makes the distinction between '(IS) design science' and. '(IS) design research. stating: "While design research is aimed at creating solutions to specific classes of relevant problems by using a rigorous construction and evaluation process, design science reflects the design research process and aims at creating standards for its rigour". Kuechler and Vaishnavi [2] have a similar view, and see DS research in the IS field as, research with design as either a topic or method of investigation.

---

[4] Wicked problems have incomplete, ill-defined requirements, contradictory, and changing requirements; solutions often difficult to recognize due to complex interdependencies. Rittel & Webber state that while solving a wicked problem, the solution of one aspect may reveal or create another, even more complex problem [18, 34].

[5] Paradigm - "the combination of research questions asked, the research methodologies allowed to answer them and the nature of the pursued research products" [6].

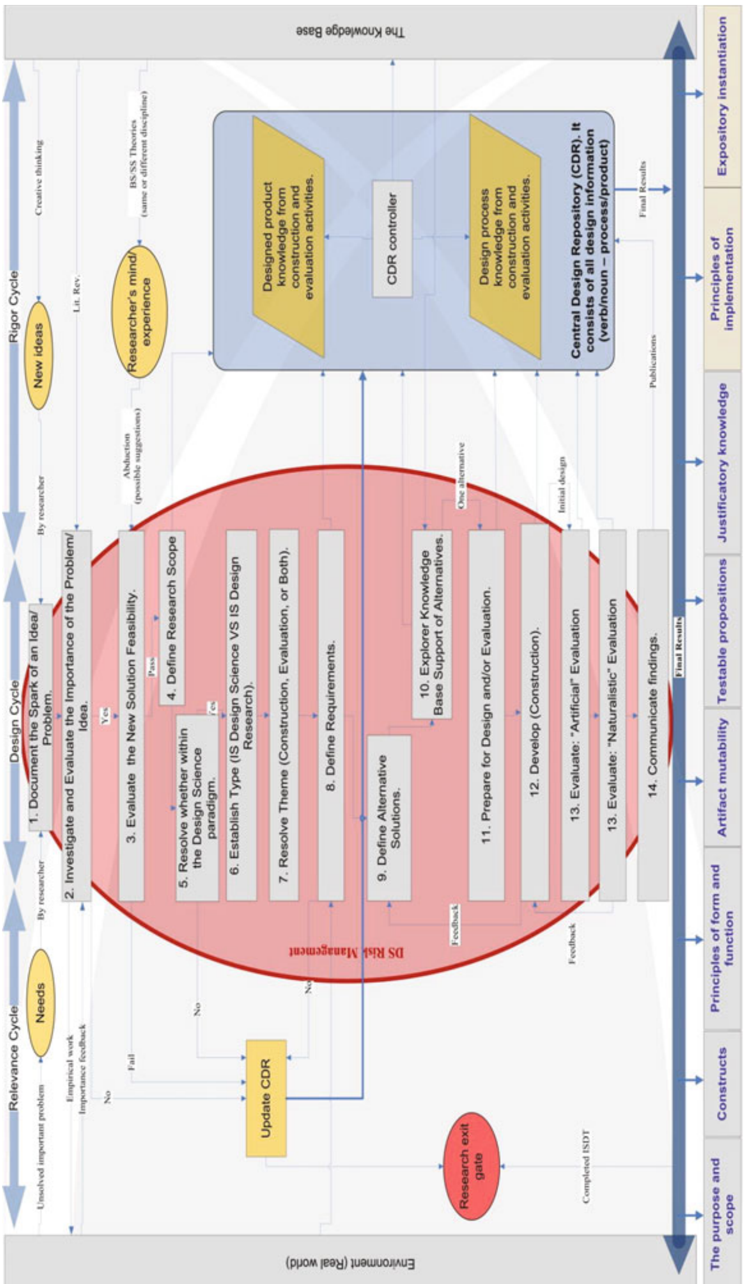


Fig. 1. The overall Design Science research Roadmap

Table 1. Design Science Activities /Steps/Tasks Distilled from the Literature

Author/Year	#	Design Science Activities/Steps/Tasks presented in the paper							
Nunamaker Jr etal.[29]	5	Construct a conceptual framework	Develop a system architecture	Analyse and design the system	Build the (prototype) system	Observe and evaluate the system			
Walls et al.[22]	7	Design Product				Design Process			
		Meta-requirements	Meta-design	Kernel theories	Testable design product hypotheses	Design method	Kernel theories	Testable design process hypotheses	
March & Smith [13]	2	Build	Evaluate						
Rossi & Sein [15]	5	Identify a need	Build	Evaluate	Learn	Theorise			
Hevner et al. [18]	7	Design as an Artifact	Problem Relevance	Design Evaluation	Research Contributions	Research Rigour	Design as a Search Process	Communication of Research	
Vaishnavi & Kuechler [16]	5	Awareness of a problem	Suggestion	Development	Evaluation	Conclusion			
Aken [39]	4	Choosing a case	Planning and implementing interventions	Reflecting on the results	Developing design knowledge to be tested and refined in subsequent cases				
Cole et al.[40]	4	Problem Definition	Intervention	Evaluation	Reflection and Learning				
Venable [17]	4	Solution technology invention	Theory building	Artificial evaluation	Naturalistic evaluation				
Peffers et al. [14]	6	Problem identification and motivation	Define the objectives for a solution	Design and development	Demonstration	Evaluation	Communication		
Gregor & Jones [25]	8	Compulsory						Optional	
		The purpose and scope	Constructs	Principles of form and function	Artifact mutability	Testable propositions	Justificatory knowledge	Principles of implementation	Expository instantiation
March & Storey [4]	6	Identification and clear description of a relevant organizational IT problem	Demonstration that no adequate solutions exist in the extant knowledge-base	Development and presentation of a novel IT artifact that addresses the problem	Rigorous evaluation of the IT artifact enabling the assessment of its utility	Articulation of the value added to the knowledge-base and to practice	Explanation of the implications for IT management and practice		
Pries-Heje et al.[31]	4	Risk identification	Risk analyzing	Risk treatment	Risk monitoring				
Pries-Heje et al. [30]	8	Ex Ante Evaluation Activity				Ex Post Evaluation Activity			
		Naturalistic Design process	Naturalistic Design product	Artificial Design process	Artificial Design product	Naturalistic Design process	Naturalistic Design product	Artificial Design process	Artificial Design product
Baskerville et al. [9]	7	A specific problem is identified and delineated	Problem must then be expressed as a specific set of requirements	The specific problem are systemically abstracted and translated into a general problem	General solution design (a class of solutions) for the general problem	General design requirements are compared with the specific problem	A declarative search is made for the specific components that will provide a workable instance of a solution to the general requirements.	An instance of the specific solution is constructed and deployed into the social system	

Goldkuhl and Lind [6] also propose a comparable distinction, dividing DS research into meta-design practice and design practice. Thus there would seem to be much consensus that this distinction is important for researchers to consider when planning and scoping their work and intended contributions.

Resolve Theme (Construction, Evaluation, or Both). Design research can entail construction, evaluation or both. March and Smith [13] believe that acceptable research need not extend to evaluation if the design solution is particularly novel. Winter [7] separates construction research from evaluation research in his IS design science research framework. Deciding on construction, evaluation, or both, is a key decision, having substantive implications for planning and activity. Evaluation may require quite different expertise and entail substantial resources.

Define Requirements. The requirements definition step specifies necessary skills, tools and experience required for the project, such as field or technical knowledge, or hardware/software resources – e.g. a specialized programming tool. To the extent possible, given the evolutionary nature of DS research, required resources and skills should be specified prior to commencing design. These requirements may be obvious, may be identified through empirical work such as interviews, surveys, case studies, etc. [25, 35] or may necessarily become apparent with the passage of time and design iteration.

Define Alternative Solutions. In essence, design is a search activity that aims to find the optimum solution to an important unsolved problem [18]. Based on Simon [42], Hevner [10] sees design as creating options that are filtered and excluded until the design's requirements are fulfilled. This step is creative, because a new solution is imagined. The defined solution is tentative and needs to be built, instantiated, and evaluated.

Explorer Knowledge Base Support of Alternatives. This step entails exploring the knowledge-base in order to discover a 'kernal' theory that supports the defined alternative solution (from Step 9). A kernel theory is a theory from Natural/Social Sciences governing design activity [22]. Kernel theories can inform design theory and at the same time, may be refined by design theory at the end of the DS research [6, 32, 43]. Gregor and Jones [25] refer to kernel theory as justificatory knowledge which is "explanatory knowledge that links goals, shapes, processes, and materials". They believe that justificatory knowledge should exist somewhere in the designed solution in DS research, even if there are limitations to this justificatory knowledge. These limitations, which are considered generated knowledge from a construction process [44], point to new areas for research [25].

Prepare for Design and/or Evaluation. This activity comprises planning for solution construction and evaluation activities. Methods for constructing the defined alternative solution are selected at this step. The step also includes preparation of functional specifications and metrics or criteria, to evaluate the significance and performance of a solution or an artifact. In this regard, March and Smith [13] believe metrics need to be defined before the evaluation process, because they play a major role in the evaluation process. Another very important task is choosing a suitable environment for the targeted solution, in which to evaluate the solution and then implement it in the real environment [39]. Moreover, since a design artifact targets an environment, including human elements, a researcher should be careful when selecting organizations and groups of people, stakeholders, when evaluating the artifact. Venable [45] identifies the stakeholders of DS research which include decision makers, professionals, future clients and others who might be impacted in the future use of design solution.

Develop (Construction). In this step, design and development of a solution for a real problem/foreseen need or novel artifact is constructed. March and Smith [13] define development as "the process of constructing an artifact for a specific purpose". A researcher tries to build an artifact that implements the alternative solution [16] from step number nine. The constructed solution, an artifact, varies in its essence based on how a researcher sees what the artifact means; there is little consensus on what exactly constitutes the artifact. Some scholars see the IT artifact as "executing code" while others view the artifact as embedded knowledge in the executing code [46]. This step also includes determination of the artifact's functionality, architecture and properties, then building an instantiation which is the physical artifact [6, 13, 14, 18, 22, 25, 47-49].

Evaluate. Once the artifact is built, it becomes the object of the evaluation activity. Rigorous evaluation often distinguishes academic from practice-based work; the evaluation process is what researchers pay close attention to and perform in academia [1]. March and Smith [13] define evaluation as "the process of determining how well the artifact performs". Hevner et al. [18] suggest evaluation "provides feedback information and a better understanding of the problem in order to improve both the quality of the product and the design process". Evaluation is an essential and empirical step because developing new/refined solution technology is still at the proposal stage and not yet proven [17]. Since the performance of the artifact depends on an intended use, the evaluation process is complex. The evaluation activity compares the performance of a solution to criteria or metrics, or functional specifications [16, 40], in the targeted environment defined in step eleven. The aim of evaluation is to decide not 'why' or 'how', but 'how well' the artifact works [13]. The new system must be verified as (1) working correctly without bugs and validated, and (2) performing required functions according to the defined requirements; these two stages are consistent with what Hevner et al. called quantitative and qualitative evaluations; and also consistent with α and β tests explained in [39], and Internal and External evaluation as in [1]. Venable [17] suggests a further, similar dichotomy - artificial and naturalistic evaluation - which we adopt in the Roadmap and discuss following:

> **"Artificial" Evaluation**. In an artificial evaluation, the designed solution or artifact is tested in a stilted way where it may pass on to external evaluation or return to the design step for refinement before entering the same loop again. If the design cannot satisfy internal evaluation, the researcher moves to another alternative solution. The design solution should only move through naturalistic, external, evaluation when it is verified and validated, because naturalistic evaluation is risky and costly.
>
> **"Naturalistic" Evaluation.** The naturalistic evaluation is the 'real' test where the invented designed solution or artifact is tested in an actual organization to check the how good or bad it is based on metrics defined in step 11. Since the

real organization is a complex structure and has many variables affecting the testing, naturalistic evaluation is difficult and may also be costly. Furthermore, as an organization has a unique combination of people, processes, etc., at any specific time, it is also not viable to compare the solution technology against different organization settings [17].

Communicate findings. This is the last step in the DS Roadmap. It means the design solution/artifact has passed the tests in the evaluation activity and can be published and communicated Most papers listed in Table 1 address this activity implicitly if not explicitly. Researchers must effectively report/communicate results, contributions, limitations, and new knowledge gained during the construction and design of the DS artifact, to communities of both researchers and practitioners; see [12, 14, 16, 20, 45]. The communication report helps practitioners to implement a solution in a new context. On the other hand, it also helps researchers to know the theoretical and methodological contributions [45]. This report is stored in a Central Design Repository (CDR) explained later, and includes all beneficial information generated across the DS lifecycle.

3.2 Output of the DS Research

This component of the DS research Roadmap is important, as researchers should, as far as possible, anticipate what they are going to produce. The authors of this paper sought to distill an inclusive DS research output taxonomy from the literature. Nonetheless, diversity of DS research outputs is observed, with little consensus [7, 16]. There is a debate on what types of output are expected from DS research [7, 16, 40, 50]. Winter [7] ponders whether it is acceptable to produce an organizational artifact, or just IT-related artifacts? Some scholars use the same output name, 'an artifact', with quite differing meanings. It was thus felt necessary to first discover all DS outputs from the literature, and then select a suitable and maximally inclusive set for the DS research Roadmap.

Table 2 summarizes DS research outputs identified from the literature review, depicting how they relate to each other, and suggesting some correspondence between scholars' conceptions, based on the authors' understanding of the related papers. This table gives a holistic view on DS research's outputs, the table rows representing key attempts to define outputs of DS research. The columns indicate the name of each output, also implying some equivalence across similar outputs suggested across the articles. A design theory as output of DS research has wide acceptance as reflected in the last column of Table 2.

Gregor and Jones's [25], the last row in Table 2, includes the widely accepted DS outputs construct, model, method and instantiation, as proposed by March and Smith [13]. They however suggest "We would argue, using authorities such as Dubin (1978) and Nagel (1979) as a reference, that "constructs, models and methods" are all one type of thing and can be equated to theory or components of theory, while instantiations are a different type of thing altogether". Their ISDT structure is most comprehensive. Moreover, Gregor and Jones's ISDT is also directly based in the work of Walls et al. [22]. On this basis, Gregor and Jones's seminal work has been adopted in the DS Roadmap.

Table 2. DS research outputs

Author	Outputs				
Nunamaker et al. [9]	--	--	--	Software	Building Theory
Walls et al.[24]	--	--	--	--	Design Theory (ISDT)
March & Smith [10]	Construct	Model	Method	Instantiation	--
Purao [42]	Operational principles			Artifact	Emergent theory
Rossi & Sein [13]	Conceptual designs	Model	Method	Systems	Better theories (theory building)
Aken [46]	--	--	--	Not considered as DS output.	Design knowledge
Venable [29]	Part of solution technology	Part of solution technology	Part of solution technology	Computer base system	Utility theory (Design Theory)
Gregor & Jones [20]	Fit into first component Of (ISDT)	Fit into a component Of (ISDT)	Fit into a component Of (ISDT)	Fit into last component Of (ISDT)	Design Theory (ISDT)

3.3 Central Design Repository (CDR)

Documenting and publishing results of research is an axiom in academe, but in DS research is supreme. Design is iterative in nature [9, 10, 14, 25, 40, 48] knowledge produced deriving from the construction process [38]. In this regard, Owen [38] states "knowledge is generated and accumulated through action. Doing something and judging the results is the general model ... the process is shown as a cycle in which knowledge is used to create works, and works are evaluated to build knowledge". Vaishnavi and Kuechler [16]] categorize the knowledge "as either 'firm' - facts that have been learned and can be repeatedly applied or behaviour that can be repeatedly invoked - or as 'loose ends' – anomalous behaviour that defies explanation and may well serve as the subject of further research". In this regard they say:

> "[T]he Circumscription in process is especially important to understanding design research because it generates understanding that could only be gained from the specific act of construction. Circumscription is a formal logical method (McCarthy, 1980) that assumes that every fragment of knowledge is valid only in certain situations. Further, the applicability of knowledge can only be determined through the detection and analysis of contradictions – in common language, the design researcher learns or discovers when things don't work "according to theory." This happens many times not due to a misunderstanding of the theory, but due to the necessarily incomplete nature of ANY knowledge base. The design process, when interrupted and forced back to Awareness of Problem in this way, contributes valuable constraint knowledge to the understanding of the always-incomplete-theories that abductively motivated the original design".

Thus, a researcher should document circumstances of all successful and failed attempts while progressing the research.

Although, most, if not all, scholars mention the significance of the documentation step in DS research and communication with academia and practice to accumulate the knowledge, no one specifies how to conduct this step, or provides any guidance to accomplish this task. In attention to this lack, this paper proposes a Central Design Repository (CDR) as an important component of the DS Roadmap. Given that an ISDT should deal with two design aspects: a designed product and a design process [22], the CDR consists of two separate parts, a designed product and a design process. The first part is document knowledge about a product such as properties, structure and functions; the latter documents the process of how to perform and implement a design solution or artifact. The CDR has a controller which will have simple criteria to be responsible for the CDR content management. The ISDT components proposed in [25] as the output of the DS research in this Roadmap are populated from the content of the CDR. This population may be gradually completed, component by component during the design progression, or at one time when the design is finished. The full content of the CDR or part of it could be published to communicate the discovered knowledge.

3.4 DS Risk Management

Risk in DS research is “a potential problem that would be detrimental to a DSR project’s success should it materialize” [31]. As depicted in Fig. 1 risk management in DS research is related to and overlaps with all steps in the DS research journey. A researcher should define, document, and monitor and be aware of every potential risk at every step in the DS research. Since there are considerable potential dangers during DS research, a researcher could prevent or mitigate these risks if s/he could predict them. Pries-Heje et al.’s [31] framework is the only work which is dedicated to address risk management in the DS research. A researcher manages these risks through four main tasks: (1) Risk identification, (2) Risk analyzing, (3) Risk treatment and (4) Risk monitoring [31]. The authors of this paper agree with Pries-Heje et al. that this framework complements the DS research methods and should be incorporated in the DS Roadmap for completeness.

4 Demonstrating the Fit of the DS Research Roadmap

In order to demonstrate the use and value of the DS research Roadmap, we apply it to two research efforts. The first is a paper, titled ‘The Design Theory Nexus’ which was published in an MIS Quarterly special issue on DS. The other is our effort to develop the DS research Roadmap itself, because the authors believe the developed DS Roadmap ‘artifact’, falls under the DS paradigm, Design Science type. In these two cases, we show how the research activities are consistent with the DS Roadmap. Language of the DS Roadmap and two simple tables have been used to demonstrate the DS research processes in these two works.

4.1 Demonstration in MISQ Case Study: The Design Theory Nexus

#	How the task is implemented
1	This research was sparked by the need in real life to solve ill-structured (wicked) problems where there is “a large degree of uncertainty with respect to how the problem should be approached and how to establish and evaluate the set of alternatives.”
2	The researchers of this case study investigated the importance of the unstructured decision making and prove that it is a wicked problem. They present how existence solutions do solve this problem; they find and prove that a decision support system was unsuitable to solve wicked problems.
3	Not mentioned or cannot be discovered.
4	The objective of this research was to “improve idealized design of problem-solving approach where a number of competing approaches exist, and where each theory has distinctive design logic.”
5	The authors of the case study argue and believe their work is under design science paradigm.
6	They did not mention anything in this regard however it can be recognized from their explanation that their research is design research DR.
7	It has been clearly seen that they construct and evaluate to produce Design Theory Nexus.
8	Building this solution “required a survey of existing literature and finding.”
9	The authors of this case study did not show what alternative they tried before they reached their solution. They perhaps omitted such discussion to avoid any type of confusion.
10	The authors of this case study support their developed solution by theory used in Carroll and Kellogg (1989). So, the solution has a kernel theory.
11	The case study articulates preparations for building a design theory nexus. It also defines how the theory will be evaluated, what evaluation metrics will be used, and who the interested stakeholders in the evaluation process are.
12	The authors of the case study design and develop five steps to build and evaluate the design theory nexus, their artifact (737-738).
13.1	To the authors’ mind of this paper, the case study did not evaluate the design theory nexus internally because the artifact based on the theory is very simple. They said “spreadsheet software proved adequate for developing the tool.”
13.2	The two instantiations, organizational change nexus and user involvement nexus, are evaluated by using real cases and 6 hr workshops, respectively.
14	The researchers communicated their finding to practitioners and researchers by publishing in a top IS journal.

4.2 Learning and Comments from Developing DS Roadmap

#	How the task is implemented
1	This effort was sparked by a need for complete steps to perform DS research. This need was discovered from very close reading of the DS literature.
2	The DS literature was investigated to check if there is a complete DS research methodology. It was found that there is a lack of a common and complete DS method and that scholars are looking for comprehensive steps for DS in IS field.
3	After gaining a deep understanding of the DS research, the development of a DS Roadmap is feasible, and resources and capabilities are available.
4	The objective of this effort is the development of a structured and general roadmap to help researchers conducting DS research.
5	The authors noticed that the process of understanding DS research and the development of the DS research Roadmap followed a DS approach. Goldkuhl [32] and Goldkuhl and Lind [6] noticed the same when they designed Multi Grounding Design Theory (MGDT) for DS research.
6	Based on Winter's [7] differentiation between design research and design science, we believe this work falls under design science because it reflects the design research process and guides artifact construction and evaluation processes.
7	The authors of this paper consider this is work as construction and relatively evaluation. The output of this effort is the DS research Roadmap.
8	Broad and deep understanding of DS research was required. The authors identified a set of articles which included most if not all attempts about DS methodology. Furthermore, software such as NVivo 8.0, EndNote, and Microsoft Visio should be used in the construction of the Roadmap.
9	The published DS methods and other discussions about DS steps that were retrieved were examined as possible alternatives for the DS Roadmap. When none proved adequate we tried iteratively to synthesize the Roadmap's steps based on these discussions.
10	The known knowledge in the knowledge-base was used to justify the Roadmap's steps. For instance, we use the knowledge-base to justify using ISDT proposed by in [25] as main component in the DS Roadmap.
11	As preparation for design and evaluation activities, the authors think that the DS Roadmap should be general, complete, moderately detailed, translatable to situated cases, and simple. The evaluation should be done by applying this roadmap on good published research as internal evaluation. External evaluation might be done later by interviewing researchers who do DS research.
12	We develop and synthesize the DS Roadmap gradually through examining every entry/code in the 'DS methods/steps' node in the Nvivo classification tree. Every single step is defined and then we grouped them under main steps. This step has been conducted and refined many times based on internal evaluation to satisfy the metrics defined at previous steps.
13.1	The roadmap was applied retroactively to published research in MISQ. Feedback was used to refine the Roadmap. We consider this internal evaluation.
13.2	An external evaluation has not been conducted yet but it could be done by interviewing researchers doing DS research to check utility, intuitiveness, parsimony, appropriate hierarchy, completeness, and how fit the DS research Roadmap with their researches and what is the value of this Roadmap.
14	The developed DS Roadmap is communicated with IS community through publishing the Roadmap in a reputable conference. This action allows researchers to benefit from this effort and evaluate it at the same time.

5 Conclusion

There is strong need for detailed guidance on conducting DS research. This paper has presented a DS research Roadmap for DS research in IS. The Roadmap is synthesized from DS-related methodological writings, providing reasonably detailed steps for researchers. The Roadmap is well grounded in literature about DS in IS. This Roadmap consists of four main interrelated components: (1) DS research cycles, (2) DS research output, (3) DS risk management, and (4) Central Design Repository (CDR). The Roadmap is general, appears to have wide application and can be translated to a situated method to carry out a specific DS research as a construction blueprint. They can also alter the situated version to fit specific design requirements. Future research may improve this Roadmap by defining lower more detailed levels of the fourteen processes and develop CDR format.

References

1. Iivari, J.: A paradigmatic analysis of information systems as a design science. Scandinavian Journal of Information Systems 19(2), 39–64 (2007)
2. Kuechler, W.L., Vaishnavi, V.K.: The emergence of design research in information systems in North America. Journal of Design Research 7(1), 1–16 (2008)
3. Samuel-Ojo, O., et al.: Meta-analysis of Design Science Research within the IS Community: Trends, Patterns, and Outcomes. In: Winter, R., Zhao, J.L., Aier, S. (eds.) DESRIST 2010. LNCS, vol. 6105, pp. 124–138. Springer, Heidelberg (2010)
4. March, S.T., Storey, V.C.: Design science in the information systems discipline: an introduction to the special issue on design science research. MIS Quarterly 32(4), 725–730 (2008)
5. Vaishnavi, V., Kuechler, W.: Design science research methods and patterns: innovating information and communication technology. Auerbach Publication, Taylor & Francis Group, NY (2007)
6. Goldkuhl, G., Lind, M.: A Multi-Grounded Design Research Process. In: Winter, R., Zhao, J.L., Aier, S. (eds.) DESRIST 2010. LNCS, vol. 6105, pp. 45–60. Springer, Heidelberg (2010)
7. Winter, R.: Design science research in Europe. European Journal of Information Systems 17(5), 470–475 (2008)
8. Iivari, J., Venable, J.: Action research and design science research–seemingly similar but decisively dissimilar. In: 17th European Conference on Information Systems (2009)
9. Baskerville, R., Pries-Heje, J., Venable, J.: Soft design science methodology. In: DESRIST 2009. ACM, Malvern (2009)
10. Hevner, A.R.: A Three Cycle View of Design Science Research. Scandinavian Journal of Information Systems 19(2), 87–92 (2007)
11. Järvinen, P.: Action research is similar to design science. Quality and Quantity 41(1), 37–54 (2007)
12. Nunamaker Jr., J.F., Minder, C., Titus, D.M.P.: Systems development in information systems research. J. Manage. Inf. Syst. 7(3), 89–106 (1991)
13. March, S.T., Smith, G.F.: Design and natural science research on information technology. Decision Support Systems 15(4), 251–266 (1995)

14. Peffers, K., et al.: A design science research methodology for information systems research. Journal of Management Information Systems 24(3), 45–77 (2007)
15. Rossi, M., Sein, M.K.: Design research workshop: a proactive research approach. Presentation delivered at IRIS 26, 9–12 (2003)
16. Vaishnavi, V., Kuechler, W.: Design research in information systems, February 20 (2004), `http://www.isworld.org/Researchdesign/drisISworld.htm` (cited January 10, 2010)
17. Venable, J.: A Framework for Design Science Research Activities. In: Information Resource Management Association Conference (CD), Washington, DC, USA. Idea Group Publishing, Hershey (2006)
18. Hevner, A.R., et al.: Design science in information systems research. MIS Quarterly 28(1), 75–106 (2004)
19. Indulska, M., Recker, J.C.: Design science in IS research: a literature analysis. In: Proceedings 4th Biennial ANU Workshop on Information Systems Foundations, Canberra, Australia (2008)
20. Venable, J.: Design Science Research Post Hevner et al.: Criteria, Standards, Guidelines, and Expectations. In: Winter, R., Zhao, J.L., Aier, S. (eds.) DESRIST 2010. LNCS, vol. 6105, pp. 109–123. Springer, Heidelberg (2010)
21. Walls, J.G., Widmeyer, G.R., El Sawy, O.A.: Assessing information system design theory in perspective: How useful was our 1992 initial rendition. Journal of Information Technology Theory and Application 6(2), 43–58 (1992)
22. Walls, J.G., Widmeyer, G.R., El Sawy, O.A.: Building an information system design theory for vigilant EIS. Information Systems Research 3(1), 36–59 (1992)
23. Campbell, D.T., Stanley, J.C., Gage, N.L.: Experimental and quasi-experimental designs for research. Houghton Mifflin, Dallas (1966)
24. Kline, R.B.: Principles and practice of structural equation modeling, 2nd edn. Guilford Press, New York (2005)
25. Gregor, S., Jones, D.: The anatomy of a design theory. Journal of the Association for Information Systems 8(5), 312–335 (2007)
26. Lee, J., Wyner, G.M., Pentland, B.T.: Process grammar as a tool for business process design. MIS Quarterly 32(4), 757–778 (2008)
27. Pries-Heje, J., Baskerville, R.: The design theory nexus. MIS Quarterly 32(4), 731–755 (2008)
28. Gregor, S., Hevner, A.: Introduction to the special issue on design science. Information Systems and E-Business Management 9(1), 1–9 (2010)
29. Nunamaker Jr, J.F., Chen, M., Purdin, T.D.M.: Systems development in information systems research. Journal of Management Information Systems 7(3), 89–106 (1991)
30. Pries-Heje, J., Baskerville, R., Venable, J.: Strategies for design science research evaluation. In: Proceedings of the 16th European Conference on Information Systems (ECIS), Galway, Ireland (2008)
31. Pries-Heje, J., Baskerville, R., Venable, J.: Evaluation Risks in Design Science Research: A Framework. In: Third International Conference on Design Science Research in Information Systems and Technology. Georgia State University, Atlanta (2008)
32. Goldkuhl, G.: Design theories in information systems-a need for multi-grounding. Journal of Information Technology Theory and Application 6(2), 59–72 (2004)
33. Venable, J.: The role of theory and theorising in Design Science research. In: Proceedings of DESRIST 2006, Claremont, CA (2006)
34. Rittel, H.W.J., Webber, M.M.: Dilemmas in a general theory of planning. Policy Sciences 4(2), 155–169 (1973)

35. Österle, H., et al.: Memorandum on design-oriented information systems research. European Journal of Information Systems 20(1), 7–10 (2010)
36. Rosemann, M., Vessey, I.: Toward improving the relevance of information systems research to practice: The role of applicability checks. MIS Quarterly 32(1), 1–22 (2008)
37. Benbasat, I., Zmud, R.W.: Empirical research in information systems: the practice of relevance. MIS Quarterly 23(1), 3–16 (1999)
38. Owen, C.L.: Design research: building the knowledge base. Design Studies 19(1), 9–20 (1998)
39. 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), 219–246 (2004)
40. Cole, R., et al.: Being proactive: where action research meets design research. In: Twenty-Sixth International Conference on Information Systems. Citeseer, Atlanta (2005)
41. Baskerville, R., et al.: A response to the design-oriented information systems research memorandum. European Journal of Information Systems 20(1), 11–15 (2010)
42. Simon, H.A.: The sciences of the artificial, 3rd edn. MIT Press, Cambridge (1996)
43. Kuechler, B., Vaishnavi, V.: On theory development in design science research: anatomy of a research project. European Journal of Information Systems 17(5), 489–504 (2008)
44. Takeda, H., et al.: Modeling design process. AI Magazine 11(4), 37 (1990)
45. Venable, J.: Identifying and Addressing Stakeholder Interests in Design Science Research: An Analysis Using Critical Systems Heuristics. In: Dhillon, G., Stahl, B.C., Baskerville, R. (eds.) CreativeSME 2009. IFIP Advances in Information and Communication Technology, vol. 301, pp. 93–112. Springer, Heidelberg (2009)
46. Baskerville, R.: What design science is not. European Journal of Information Systems 17(5), 441–443 (2008)
47. Järvinen, P.: On reviewing of results in design research. In: The 15th European Conference on Information Systems, Switzerland (2007)
48. Purao, S.: Design research in the technology of information systems: Truth or dare, Unpublished Working Paper, Atlanta (2002)
49. Orlikowski, W.J., Iacono, C.S.: Desperately seeking the'IT'in IT research-a call to theorizing the IT artifact. Information Systems Research 12(2), 121–134 (2001)
50. Offermann, P., et al.: Artifact Types in Information Systems Design Science – A Literature Review. In: Winter, R., Zhao, J.L., Aier, S. (eds.) DESRIST 2010. LNCS, vol. 6105, pp. 77–92. Springer, Heidelberg (2010)

Discovering the Meanings of Design in IS: Reviews and Future Directions

Jaehyun Park[1], Richard Boland Jr.[1], and Youngjin Yoo[2]

[1] Information Systems, Weatherhead School of Management,
Case Western Reserve University, Cleveland, OH 44106, U.S.A.
{jaehyun.park,boland}@case.edu
[2] Management Information Systems, Fox School of Business, Temple University,
Philadelphia, PA 19122, U.S.A.
youngjin.yoo@temple.edu

Abstract. The purpose of this study is to identify the meanings of design to Information Systems (IS) scholars as revealed in their research. We conducted an extensive review of IS research papers from 1970 to 2007 that are related to design research broadly construed and analyzed them to reveal the overview of their scope and patterns using two approaches. The first approach locates existing research in a design space based on attention to design artifacts versus design processes, and their interests in the analytic or the synthetic aspect of design cognition. The second approach locates existing research in a design space based on the level of organization system from local to global that design affects, and the temporal state (past, present, future) that it considers. Finally, this study draws upon where the lack of design research in IS remains as future research opportunities.

Keywords: design research in IS, action research, design research framework.

1 Introduction

Design has always been a central issue in Information Systems (IS) research and practice, and IS design domain is expanding its concerns from fixed system issues to open social ones, encompassing IT, media, and digital solutions. Here, design has been treated as an essential part of the development and implementation of information systems, and IS scholars need to pay more attention to design. Offering an extensive perspective, Hirschheim et al. (1995) provide a conceptual and philosophical foundation of the boundary of information system design. Orlikowski and Iacono (2001) call for more careful research on IT artifacts. Hevner et al. (2004) affirm design science as an important intellectual branch in the IS community. Prior IS design science scholars have theoretically explored what design science is, what design issues are useful in IS contexts, and why design is an important discipline in ISD (Information System Development) and IS artifacts in few journal special issues (e.g. SJIS 2007 Issue 2, EJIS 2008 Issue 5, and MISQ 2008 Issue 4) and conferences (e.g. DERIST, AMCIS mini-tracks). These studies have significantly expanded the

H. Jain, A.P. Sinha, and P. Vitharana (Eds.): DESRIST 2011, LNCS 6629, pp. 124–137, 2011.

boundaries and the theoretical visions of IS design science research over time. Yet, we believe that design has already been embedded into IS research and in the work of prior IS researchers over time, not just in design science research as a new IS discipline.

In this paper, we discover how the published IS research have developed the meanings of design and where the researchers design considerations exist. The purpose of this study is to identify new research opportunities for the IS community by reflecting on the broad nature of design and published IS design related studies. In what follows, we conduct two reviews with different perspectives to elucidate the science of design in IS: a designer's perspective in the first review, and then a design researcher's perspective in the second review.

The first review examines research from a designer's standpoint, theorizing two dimensions of design: what things are considered (design action) and how things are considered (design cognition). We address the question -- What design action and design cognition have IS researchers been concerned with in IS research? To do that, we develop a theoretical framework for reviewing IS publications by design action and design cognition. The design action may be the processes (ways of creating a design) or outcomes (things being created). Design cognition ranges from analysis and synthesis in varying degrees. With this analytic framework, we conduct a content analysis of published IS research and classify the attention to design action and design cognition in IS over time.

Our second review explores research from an IS researcher's standpoint, identifying design research spaces with two design dimensions: what scope of design is considered and what design timeframe is considered. Here, our guiding question is -- What aspects of IS design have been studied by IS researchers? In order to address this, we perform a grounded analysis approach to identify two dimensions reflected in the published IS research: the scope of design being considered and the timeframe of the IS design being studied. The scope of design ranges from particularities to the ecological wholeness of a system. The timeframe being considered ranges from previously constructed (existing system designs), to currently being constructed (emerging design systems), to future constructions (design systems that are planned for the future). We refer to a combination of scope and timeframe as an IS research space, and locate each study in a research space.

2 Methodology

We sampled two premier IS journals, *MIS Quarterly* (MISQ) and *Information Systems Research* (ISR) because these two journals deal with major IS research, and most leading scholars' works are included. In order to filter design articles broadly related to the nature of design in MISQ and ISR, we listed all papers from the initial issues to the year of 2007, excluding editorial notes and interview sections. Subsequently, we printed all of the abstracts of MIS Quarterly and ISR over those time periods.

We performed two stages of sampling process to filter design papers in IS (see Table 1 for the summary). In the first stage, 1,132 *abstracts* are reviewed. Here, we selected 252 papers as design papers and rejected 880. In the second stage, we reviewed the 252 *full* *papers* that were deemed as design papers from the first stage. Based on the full text reading, we identified 162 papers as design related papers (MISQ: 106, ISR: 56), and these 162 papers became the basis for our analysis.

Table 1. The Process of Sampling

	The 1st Paper Selection with Abstracts			The 2nd Paper Selection with Full Papers		
	Design Papers	Not Design Papers	Total	Design Papers	Not Design Papers	Total
MIS Quarterly	169	597	766	106	63	169
ISR	83	283	366	56	27	83
Total	252	880	1,132	162	90	252

We developed two analytic frameworks to review the sample papers. The objective of the first review is to account for what design topics exist in IS. To elucidate this, we created a framework, which deals with design action and design cognition as two dimensions from a designer's perspective by using a content analysis across the 162 presented design studies in IS. On the other hand, the objective of the second review is to locate the 162 presented design research papers by a grounded theory approach. The two dimensions from IS researchers' perspective are comprised of the scope of design being considered and the design timeframe of the researcher toward design action. The detailed review process will be explained in each review section (section 3 and section 4).

3 Design Action and Cognition in IS Research: A Designer's Perspective

3.1 Empirical Approach: Content Analysis by Analytic Framework

Drawing upon the extant literature on design, our first analytical framework is based on two design dimensions shown in Figure 1: *design action* and *design cognition*. Design action can be either the process of designing or the product of design (Boland 1978). On the other hand, design cognition is analyzing or synthesizing in varying degrees. These two design dimensions are used to identify *what design is* from a designer's standpoint (Yoo, Boland, & Lyytinen, 2006).

As the dimensions of design action, *design as artifacts* represents design-as-noun (the things being created), and includes the forms of 2D (images), 3D (products), interaction / system designs, and service designs, while *design as processes* accounts for design-as-verb (the processes of creating things), and it encompasses the processes of information processing, effective prototypes, coordination, learning and knowledge development, and so on. In design cognition, *design as analyzing* deals with evaluations, feedbacks, systemic analysis for design requirements, resources,

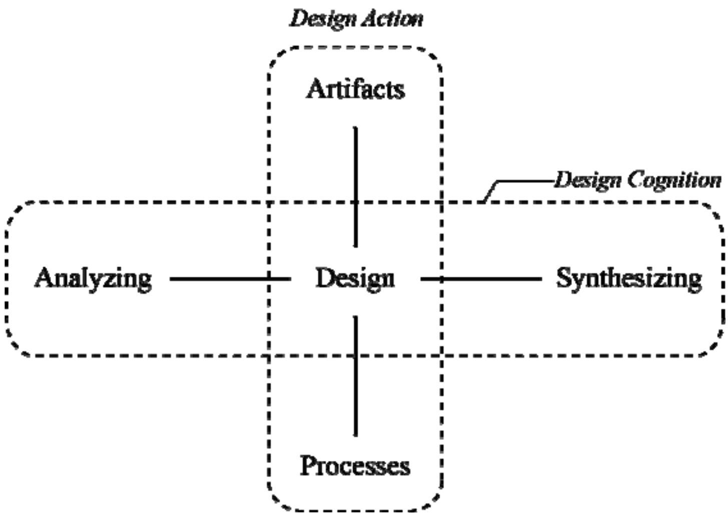


Fig. 1. The First Framework: A Viewing the Designer's Action and Cognition

behaviors, materials, and alternatives, while *design as synthesizing* combines the synthesis of new ideas, forms, systemic tasks and functions, and relationships of actions in system designs. Taken together, we review past IS research broadly related to design action and design cognition with four design categories in Figure 1: Synthesizing Artifacts (SA), Analyzing Artifacts (AA), Analyzing Processes (AP), and Synthesizing Processes (SP).

In order to perform a systemic content analysis (Duriau et al 2007, Pentland 1999), we recursively categorized the 162 papers according to the two main dimensions (design action and design cognition) by classifying them into one of the four design categories, and then we reclassified 19 sub-categories as 19 IS design topics.

3.2 Result and Findings in the First Review

Figure 2 represents 19 design topics over four design categories, and if required, we defined sub-categories among the 19 design topics (e.g. design elements and application & platform).

As extreme cases of each design category, we discuss four topics: IT artifacts in SA, usability in AA, learning and training in AP, and user-centered design in SP. IT artifacts in SA account for synthesizing IT solutions or outcomes for individual, groups, and organizations. The topic combines Electronic Meeting System (EMS), Electronic Brain Storming (EBS), Creativity Support System (CSS), and Knowledge Based System (KBS). Usability in AA deals with analyzing emerging design applications or systems such as software usability, design interface, and usability guidelines. Learning and training in AP deals with analyzing the processes of end-user learning and the effective procedural mechanisms in software training. User-centered design in SP accounts for the impact of user-centric approaches defined by the following concerns: how user generated information systems theorize new principles of ISD, and how user centered design synthesizes new materials of IT artifacts. Among the 19 design topics, standards and guidelines and methods and models occupy two liminal positions, because they are less extreme and therefore more difficult to classify. Here, we identified the standards and guidelines of SA, because they are a form of DSS, managerial strategies, user oriented systems, and optimal software development, etc. Also, we defined methods and models as the category of SA, to evolve new types of artifacts because of its diverse methodologies of strategic design and linguistic approaches in IS.

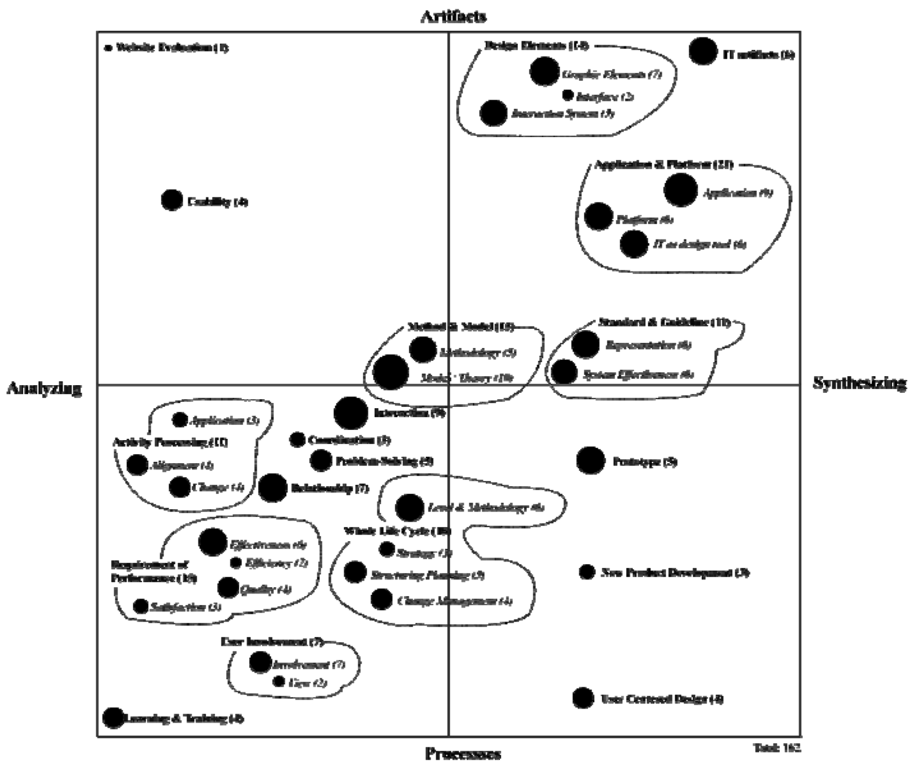


Fig. 2. The Result of the First Review

As the interpretation of the first review, Synthesizing Artifacts (SA) and Analyzing Processes (AP) occupy most design topics in IS, while Analyzing Artifacts (AA) and Synthesizing Process (SP) have not been adequately studied. This result demonstrates that most IS design studies can be categorized either SA or AP, and it reflects the research pattern of IS design from a designer' view. Here, the published papers focus on what IS designers' actions have to be analyzed to identify a design system, and what design systems can be synthesized through the analytic designers' actions. Considering the reasons for the lack of AA and SP research in IS design, the AA and SP illustrate two design issues from a designer' view: (1) how designers analyze design systems and design actions; and (2) how new actions and alternative methods support design systems and design actions. Therefore, recent studies have considered AA and SP categories as alternative research areas to expand new design topics.

Looking at the design topics closely, the 19 topics acquire dynamic meanings across each topic. For example, the topic of design elements falls under IT artifacts as the aspect of applications, and the topic of IT artifacts applies to application and platform as a part of IT infrastructures. Also, design elements, IT artifacts, and application and platform also generate a meaning – the attention to forms and functions in IS. In this way, all 19 design topics are connected one to the other, and they might generate new classifications for the science of design in IS. Therefore, we might say the 19 design topics draw upon the picture of design in IS as an answer to *what design is in IS*. The 19 design topics are a set of useful constructs to understand what IS researchers reflect in terms of the views of design action (design as artifacts or design as processes) and design cognition (design as analyzing or design as synthesizing).

Table 2. Four Design Categories in IS over Time

	1970s	1980s	1990s			2000s			Total		
	MISQ	MISQ	MISQ	ISR	Sub Total	MISQ	ISR	Sub Total	MISQ	ISR	Total
Synthesizing Artifacts (SA)	1.2% (2)	8.6% (14)	**9.2% (15)**	**6.8% (11)**	16.0% (26)	**7.4% (12)**	**8.0% (13)**	15.4% (25)	**26.5% (43)**	**14.8% (24)**	**41.3% (67)**
Analyzing Artifacts (AA)	0% (0)	0% (0)	1.2% (2)	0% (0)	1.2% (2)	0% (0)	1.8% (3)	1.8% (3)	1.2% (2)	1.8% (3)	3.0% (5)
Analyzing Processes (AP)	5.5% (9)	16.0% (26)	**6.8% (11)**	**9.2% (15)**	16.0% (26)	**4.3% (7)**	**6.1% (10)**	10.4% (17)	**32.7% (53)**	**15.4% (25)**	**48.1% (78)**
Synthesizing Processes (SP)	0.6% (1)	2.4% (4)	1.2% (2)	0.6% (1)	1.8% (3)	0.6% (1)	1.8% (3)	2.4% (4)	4.9% (8)	2.4% (4)	7.4% (12)
Total	7.3% (12)	27.0% (44)	18.4% (30)	16.6% (27)	35.0% (57)	12.3% (20)	17.7% (29)	30.0% (49)	65.5% (106)	34.5% (56)	100% (162)

Total: 100% (N=162)

Table 2 represents the result of the coding from the four design categories in IS over time. Synthesizing Artifacts (SA) and Analyzing Processes (AP) (89.4%) have dominated the view in IS design research since the 1970s; while Analyzing Artifacts (AA) and Synthesizing Processes (SP) (10.4%) have not developed adequately in IS research; however, both areas have been expanded since the 1990s. In addition, MISQ and ISR have different concentrations of four design categories. MISQ has highlighted the Analyzing Processes (AP) more than ISR, while ISR has focused on the Synthesizing Artifacts (SA) more than MISQ. When we consider the fact that the initial issue of ISR was in 1990, MISQ has more Synthesizing Artifacts (SA) research (9.2% in 1990s, and 7.4% in 2000s) than Analyzing Processes (AP) (6.8% in 1990s, and 4.3% in 2000s). On the other hand, ISR has had similar research volumes in Synthesizing Artifacts (SA) (6.8%, and 8.0%) and Analyzing Processes (AP) (9.2%, and 6.1%) since the 1990s.

4 Design Research Spaces in IS Research: A View from an IS Researcher's Perspective

4.1 Empirical Approach: Analytic Framework by Grounded Analysis Approach

In the second review, we used a grounded analysis approach (Corbin & Strauss 1990, Glaser & Strauss 1967) across the published IS design papers, because there is no prior theoretical framework that we can draw on. We reanalyzed the sample asking a following question – *what aspect of IS design is being subjected to research?* Table 3 shows the result of our iterative coding process in the second review. Our grounded analysis approach reveals two systems that underpin prior design related research in IS. One system (X-axis) deals with *design timeframe* and it indentifies three scales from past to future system action as follows: (1) existing system designs, (2) emerging system designs, and (3) future system designs. The other system (Y-axis) deals with *the scope of IS design.* It defines five levels ranging from *the most particularly detailed* (the aspects of an application) to *the largest whole* (the ecology of the system): (1) aspects of application, (2) single application, (3) infrastructure, (4) system development, and (5) the whole ecology of system.

4.2 Result and Findings in the Second Review

Table 3 represents the result of the second review developed through a grounded analysis approach that codes the 162 presented design studies. Here, we identify 15 design research spaces for the science of design in IS.

Through Table 4, we summarize the result of second review and elucidate *three* insights.

First, considering the volume of research across 15 design research spaces, design research in IS has been well balanced. In design timeframe (x-axis), the system designs have been dealt with as follows: existing system designs is 15.4%, emerging system designs is 42.0%, and future system designs is 42.6%. Meanwhile, regarding the scope of design (y-axis), the system designs have been dealt with as follows: aspect of application is 15.4%, application is 27.2%, infrastructure is 11.1%, system development is 24.7%, and the whole ecology of system is 21.6%.

Second, looking at each of the 15 design research spaces, application (7.4%) in existing system designs, aspects of application (9.3%), application (11.1%), infrastructure (8.6%), and system development (8.6%) in emerging system designs, application (8.6%), system development (11.7%), and the whole ecology of system (21.6%) in future system designs are the most popular design research spaces. The patterns of popularity revealed above demonstrate that the design research in IS has focused on emerging system designs in order to identify unexamined theories and future system designs. This would allow researchers to take design research in a direction whereby they could develop the quality of IT artifacts and system design action.

Third, MISQ and ISR have different concentrations. In comparing MISQ and ISR, MISQ has focused on emerging system designs (23.5%) and future system designs (34.6%), while ISR has concentrated more on existing system designs (8.6%) and emerging system designs (18.5%) in design timeframe (x-axis). In the scope of design (y-axis), MISQ has highlighted application (19.2%), system development (17.3%), and the whole ecology of system (15.4%). On the other hand, ISD has been balanced on each design research space (6.2%, 8.0%, 7.4%, and 6.2%). In MISQ, the most popular design research spaces are aspects of application (7.4%), application (6.2%) in emerging system designs and application (8.0%), system development (10.5%), and the whole ecology of system (12.4%) in future system designs. On the other hand, ISR publishes work on application (4.9%), infrastructure (6.2%), and system development (3.7%) in emerging system designs. These results allow us to conclude that ISR is more focused on discovering emerging system designs and applications so that authors of ISR conducted more empirical studies with quantitative evidence; however, MISQ has more higlighted to future system designs with diverse approahces so that the scholars in MISQ performed more theoretical, case studies by qualitative evidence.

Table 3. 15 Design Research Spaces in IS

	Existing System Designs (1. 0)	**Emerging System Designs (2. 0)**	**Future System Designs (3. 0)**
Aspects of Application (0. 1)	**Space 1: (1. 1)** Goldstein & Storey (1992) Clifford et al. (1996)	**Space 6: (2. 1)** Suh & Jenkins (1992) Kim et al. (2000) Hong et al. (2004) Firth (1980) Vassiliou et al. (1983) Watson & Driver (1983) Benbasat & Dexter (1986) Lee & MacLachlan (1986) Olfman & Mandviwalla (1994) Tractinsky & Meyer (1999) Lim & Benbasat (2000) Speier & Morris (2003) Kumar & Benbasat (2004) Suh & Lee (2005) Nadkarni & Gupta (2007)	**Space 11: (3. 1)** Mookerjee & Santos (1993) Schocken&Christopher(1993) Mannino et al. (1994) Krishnan et al (2001) Aalst & Kumar (2003) Ives (1982) Banker & Kauffman (1991) Orlikowski (1993)
Application (0. 2)	**Space 2: (1. 2)** Mackay & Elam (1992) Pinsonneault et al. (1999) Dennis & Valacich (1999) Agarwal &Venkatesh (2002) Tan & Benbasat (1990) Schwartz et al. (1980) Semprevivo (1980) Sprague (1980) Alavi (1981) Chandler (1982) Rivard & Huff (1984) Wetherbe (1991) Storey & Goldstein (1993)	**Space 7: (2. 2)** Prietula &March (1991) McLeod & Liker (1992) Sambamurthy & Poole (1992) Robey & Sahay (1996) Pinsonneault et al. (1999) Palmer (2002) Lilien et al. (2004) Kumar & Benbasat (2006) Olson &Ives (1982) Lambert & Wallace (1990) Chan et al. (1993) Lee (1994) Massetti (1996) Webster (1998) Wierenga & Bruggen (1998) Mennecke et al. (2000) Sircar et al. (2001) Venkatesh & Ramesh (2006)	**Space 12: (3. 2)** De et al. (1993) Alter (1978) Fowler (1979) Locander, et al. (1979) Barbosa & Hirko (1980), Vierck (1981) Huber (1984) Henderson&Schilling (1985) Borovits & Neumann (1988) Mantel & Teorey (1989) Apte et al. (1990) Growwski et al. (1990) Albert et al. (2004) Hirshheim (1985)

Table 3. (*continued*)

Infrastructure (0. 3)	**Space 3: (1. 3)** Sinha & May (1996)	**Space 8: (2. 3)** Limayem & DeSanctis (2000) Nault & Vandenbosch (2000) Banker & Slaughter (2000) McKinney et al. (2002) Asvanund et al. (2004) Jones et al. (2004) Jiang et al. (2005) Dellarocas (2005) Tam & Ho (2005) Gu et al. (2007) Sawy & Bowles (1997) Broadbent et al. (1999) Tam & Ho (2006) Webster & Ahuja (2006)	**Space 13: (3. 3)** De &, Sen (1984) Trauth & Cole (1992) Palvia et al. (1992)
System Development (0. 4)	**Space 4: (1. 4)** Gremillion (1980) Olson & Ives (1982) Janson & Smith (1985) Koushik&Mookerjee(1995) Medor & Mezger (1984) Gerlarch & Kuo (1991) Nambisan (2003)	**Space 9: (2. 4)** Elam & Mead (1990) Weber (1996) Garfield et al. (2001) Purao et al. (2003) Chiang & Mookerjee (2004) Banker et al. (2006) Moore (1979) Kaiser & Bostrom (1982) Tait & Vessey (1988) Dekleva (1992) Aherens & Sankar (1993) Davis & Bostrom (1993) Lawrence & Low (1993) Barki & Hartwick (1994)	**Space 14: (3. 4)** Ji et al. (2005 Kasper (1996) Kling (1977) Benbasat & Taylor (1978) Zmud, & Cox (1979) Gremillion (980) Fuerst & Martin (1984) Mann & Watson (1984) Janson & Smith (1985) Barons & Louis (1988) Jessup et al. (1990) Karimi (1990) Te'eni (2001) Lindgren et al. (2004) Slaughter et al. (2006) Halloran et al.(1978) McLean (1979) Baskerville & Stage (1996) Markus et al. (2002)
The Whole Ecology of System (0. 5)	**Space 5: (1.5)** Sillince&Mouakket (1997) Iivari,et al.(1998) King (1982)	**Space 10: (2. 5)** Wright et al (1998) Marakas & Elam (1998) Lilien et al. (2004) Benbasat, et al. (1977) White & Leifer (1986) Mahmood (1987) Necco et al. (198 7)	**Space 15: (3. 5)** Kirsch (1997) Hirschheim & Newman (1991) Walls et al (1992) Ba et al. (2001) Levina (2005) Ahituv et al. (1984) Doll (1985) Meador & Rosenfeld (1986) Kozar & Mahlum (1987) Sherif &Sawy (1988) Couger et al. (1993) Clark et al. (1997) Nissen (1998) Ravichandran, & Rai (2000) Hevner et al. (2004) Highsmith (1978) Berrisford & Wetherbe (1979) Schonberger (1980) Naumann, & Jenkins (1982) Roby & Markus (1984) Zmud (1984) Lyytinen (1985) Lovata (1987) Kendall & Kendall (1993) Hirshheim & Klein (1994)

Table 4. The Result of the Second Review

	Existing System Designs	Emerging System Designs	Future System Designs	Total % N)
Aspects of Application	1.2% (2)	**9.3%** (15)	4.8% (8)	15.4% (25)
Application	**7.4%** (12)	**11.1%** (18)	**8.6% (14)**	**27.2% (44)**
Infrastructure	0.6%(1)	**8.6% (14)**	1.8% (3)	11.1% (18)
System Development	4.2%(7)	**8.6% (14)**	**11.7%** (19)	**24.7% (40)**
The Whole Ecology of System	1.8% (3)	4.3%(7)	**15.4%** (25)	**21.6% (35)**
Total % N)	15.4% (25)	**42.0%** (68)	**42.6%** (69)	100% (162)

5 Future Research Opportunities

5.1 Research Opportunities in the First Review

Figure 3 represents the result of the first review regarding what IS research has been considered from the designer's perspective. As a result, it describes the Synthesizing Artifacts (SA) and Analyzing Processes (AP) dominating IS design research, while Analyzing Artifacts (AA) and Synthesizing Processes (SP) are sparse design research areas.

The AA research has dealt with the topics of design feedback, usability testing, and applicable evaluation methods. The SP has also dealt with the topics of prototyping, NPD, and UCD as alternative system implementation; however, the potential to create design topics into AA and SP research are limited.

In order to overcome sparse research areas in AA and SP, we ask the following question: *what aspects of the SA and AP can produce new design topics in the AA and SP*. We then propose four research paths: (1) SA → AA, (2) SA → SP, (3) AP → AA, and (4) AP → SP.

First, the path from SA to AA demonstrates new types of analytic tools in AA research. In IT convergence era, this path considers analytical methods to create new IS design systems and their embedded services by constructing and reconstructing current IT system designs. For instance, the eye-tracking technique of new product development shows where design problems exist in emerging design artifacts, and it makes design process logical for identifying reliable IT system designs. The reputation mechanism of Amazon and eBay also represent how analysis tools could be embedded into existing or emerging IT artifacts. In addition, the personalized advertising service in Facebook, Google, and LinkedIn represent what current artifacts require new analytic forms, functions, and services as potential AA research.

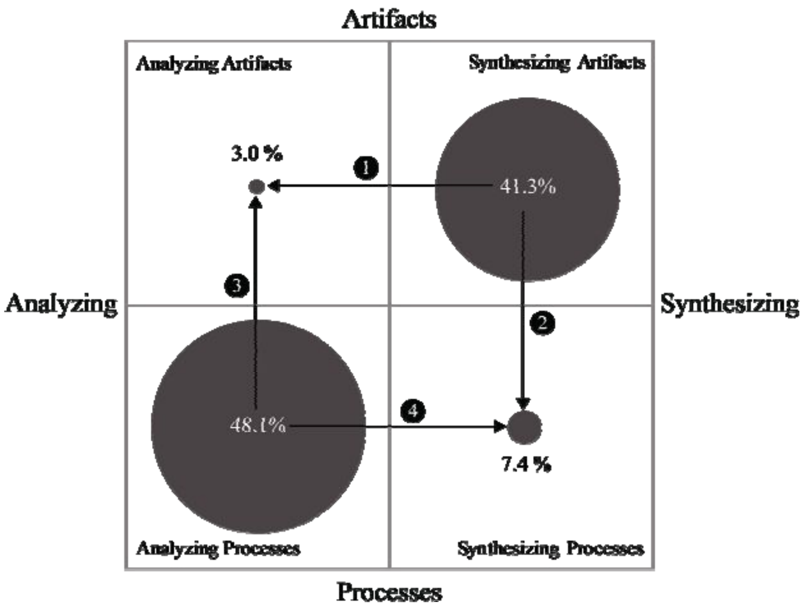


Fig. 3. Result of the First Review

Second, the path from SA to SP explains two future research topics: (1) ethnographic approaches, and (2) co-creation among multi-stakeholders. The ethnographic approaches characterize new methods and techniques in order to understand the requirements of a design process to create new system designs such as Apple iTunes and iLife service, FedEx overnight delivery service, and mobile software packages within USB. The co-creation among multi-stakeholders will be an important research topic to identify new types of design activities, and Wikipedia, online forums, Google dot, and Blogs present how the co-creation among multi-stakeholders can generate design research topics in SP.

Third, the path from AP to AA deals with the aspects of AP that require new analytic methods of AA to test design actions. For example, the interaction of AP need information architect as an analytic method to overview the workflow of existing / emerging design systems. The analytical techniques of AA support effectiveness, efficiency, quality, and satisfaction of AP.

Fourth, the path from AP to SP accounts for what aspects of design action in AP characterize synthetic design action in SP. For example, the whole life cycle of AP requires not only analyzing diverse goals, roles, and tasks of system actors, but also synthesizing actions in order to expand more dynamic interactions in emerging / future system designs. In doing so, the analytic actions of the whole life cycle can shape synthetic design actions in SP. The user centered design approach, new product development skills, and prototypes are applicable actions to connect the actions between AP and SP.

5.2 Research Opportunities in the Second Review

The 15 design research spaces from a researcher's perspective show more balanced than the first review in Figure 4; however, there are still less developed research areas. In order to overcome some less developed research areas in the second analysis, we see three research potential that have not been studied much by prior researchers: (1) aspects of application, (2) infrastructure, and (3) existing system designs.

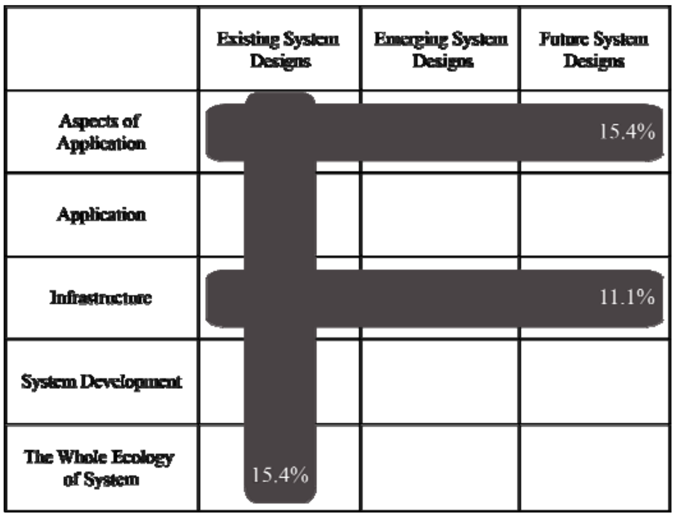


Fig. 4. The Result of the Second Review

First, the aspects of application research accounts for the impact of IT. Prior studies have dealt with the effects of a single IT technology and design elements, leaving out the broader issue of the nature of IT and its design elements. Future research could attempt to understand specific forms, functions, and embedded technologies in the aspects of application, and then to identify where the important aspects of particularities exist in IS system designs. User Interface (UI) will be an example of possible study. A successful UI design requires a combination of the units of human factors and design components. Thus, a UI design is made up of Graphic User Interface (GUI), Physical User Interface (PUI), Tangible User Interface (TUI), Audible User Interface (AUI), and information architecture. Also, as a part of UI, a successful GUI design requires a harmony between the resource of graphic design and human behaviors. Thus, a GUI design consists of intuitive metaphors, easy navigations, and reliable orientations. As a particularity of GUI, a successful navigation design requires a clear definition of events, contents, contexts and so on. Thus the importance of particularities will offer multi-layered design research opportunities to the aspect of application.

Second, the research on infrastructure accounts for an outcome of socio-technologies, and a successful infrastructure requires a combination from design particularities to ecological wholeness in a system design. Yet, prior infrastructure research has not been studied well, because it is associated with design policy. Thus, we highlight design policy as future research opportunities.

Design policy is a central issue for realizing ubiquitous cities. In many cases, design policy has produced new design research opportunities such as urban planning, the economic values by establishing technologies, and the patterns of urban experience. In 2010, Seoul, South Korea became the world design capital, illustrating how the design policy can create new social infrastructure. There are two design principles defining a design capital: (1) material-driven IT infrastructure by high technology, consumer products, and the spirit of change in social movements; and (2) government-driven scenario planning. In this regard, design policy will locate infrastructure research opportunities.

Third, the research of existing system designs has significantly contributed to understanding design spaces; however, these coded design research spaces on existing system designs have not been studied in terms of the history of technology. Thus, here remains a design research opportunity for how the history of technology can support new design research spaces in institutions or socially constructed usages of technology.

Recently, cloud computing has become a central issue in terms of future computing applications. People exploit its computing utility and economic value for synthesizing new types of behaviors such as accessing limitless computing power without the barriers of terminal devices as in a pay-as-you-go arrangement; however, the benefits of cloud computing have also understood in terms of previous computing technologies such as database, internet, mobile, or physical computing. Therefore, the history of technology of existing design systems can characterize new research opportunities in IS design research.

6 Conclusions

In this paper, we have sought to discover design spaces in IS and to identify opportunities for future design research in IS. Like any research, this study also has several limitations. First, we only reviewed MISQ and ISR. It is possible that inclusion of diverse journals (for example European journals or those that emphasize design science) and practitioners' endeavors could identify a wider range of meanings for the science of design in IS. Second, the development of our analytical research framework is based on a subjective way of identifying the body of knowledge for design in IS. Others might conduct a similar study based on different dimensions that reflect the diverse nature of design. Third, IS design research is an emergent method, so that our frameworks and findings could be interpreted differently by different perspectives on the same studies.

Despite of these shortcomings, we believe our study provides valuable insights to IS researchers who are interested in design. We encourage IS researchers to consider new design research directions in order to expand their current research topics, and we further encourage IS researchers associated with design to broaden their ranges of inquiry by considering more diverse approaches to both empirical and theoretical research.

References

1. Boland Jr, R.J.: The process and product of system design. Management Science 24(9), 887 (1978)
2. Corbin, J.M., Strauss, A.: Grounded theory research: Procedures, canons, and evaluative criteria Qualitative Sociology 13(1) (March 1990)
3. Duriau, V.J., Regar, R.K., Pfarrer, M.D.: A content analysis of the content analysis literature in organizational studies. Organizational Research Methods 10, 5–37 (2007)
4. Glaser, B.G., Strauss, A.L.: The Discovery of Grounded Theory: Strategies for a Qualitative Research. Aldine Publishing Company, Chicago (1967)
5. Pentland, B.T.: Building Process Theory with Narrative: From Description to Explanation. Academy of Management Review 24(4), 711–725 (1999)
6. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28(1), 75–105 (2004)
7. Hirschheim, R., Klein, H.K., Lyytinen, K.: Information Systems Development and Data Modeling: Conceptual and Philosophical Foundations. Cambridge University Press, Cambridge (1995)
8. Orlikowski, W.J., Iacono, C.S.: Research Commentary: Desperately Seeking the 'IT' in IT Research - A Call to Theorizing the IT Artifact. Information Systems Research 12(2), 121 (2001)
9. Yoo, Y., Boland, R.J., Lyytinen, K.: From organization design to organization designing. Organization Science 17(2), 215–229 (2006)

Design and Behavioral Science Research in Premier IS Journals: Evidence from Database Management Research

Anteneh Ayanso[1], Kevin Lertwachara[2], and Francine Vachon[1]

[1] Faculty of Business, Brock University, 500 Glenridge, St. Catharines, ON, L2S 3A1 Canada
[2] Orfalea College of Business, Cal Poly State University, San Luis Obispo, CA 93407
{ayanso,fvachon}@brocku.ca, klertwac@calpoly.edu

Abstract. In this article, we examine database management research that has been published in ISR, JMIS, and MISQ from each journal's inception to 2007. Our goal is to profile database research using a classification scheme that includes research paradigms, IT constructs, and research methodologies. The overall statistics obtained shows that information systems (IS) research in database management, which is widely recognized as part of the core knowledge of IS, is diverse in IT constructs, methodologies, as well as research paradigms. However, we also find that each journal has focused more on one research paradigm and some research methodologies. We summarize and discuss these results which can be useful to design science researchers in targeting their work in these three premier IS journals.

Keywords: Design Science, Behavioral Science, IS Research, Database Research, Research Diversity.

1 Introduction

Much of the long-standing debate over the nature of Information Systems (IS) focuses on whether or not research diversity in our field is desirable [6, 7]. Past studies have examined various aspects of diversity in our field including diversity in reference disciplines, research topics, and methodologies [3]. Among these discussions on IS research diversity, the issue of design versus behavioral science has recently gained much attention in our community. Prominent IS researchers have argued for more focus on design science research and for greater recognition of design science research among premier IS journals (see, for example [22] and [31]). The seminal work by Hevner et al. [22] argues that the design and behavioral science paradigms should co-exist alongside and even complement one another.

Since its inception, research in IS has drawn upon a "bewildering variety" of theoretical foundations and methodologies [40]. The pluralistic nature of our field is a consequence of our "varied origins" as founders of our field came from backgrounds that include Computer Science, Economics, Management Science, Physics, and Psychology [21]. This pluralism is our heritage and influences how we, as an academic field, have come to define ourselves [38]. At the first international

H. Jain, A.P. Sinha, and P. Vitharana (Eds.): DESRIST 2011, LNCS 6629, pp. 138–152, 2011.

conference among IS scholars, for example, Keen [25] defined IS as "a fusion of behavioral, technical, and managerial issues." In his editorial statement in ISR, King [26] described IS as an "intellectual convocation of individuals from many fields…" In a more recent example, Galliers [17] not only recognizes that the "roots" of IS are found in a variety of reference disciplines, but also advocates that we should strive to become even more "trans-disciplinary." This "trans-disciplinary" ideal was also echoed by Myers [32] (as quoted in [30]) as a collaborative environment where IS scholars from different research perspectives and approaches work together "within the scope of a single research project or within a particular research area."

The focus of this study is on database research which is widely recognized as a quintessential part of the IS discipline [10, 34, 45]. We explore the diversity in database management research that has been published in the top three 'mainstream' IS journals. Specifically, we examine all database research articles that have been published in *Information Systems Research (ISR). Journal of Management Information Systems (JMIS)*, and *MIS Quarterly (MISQ)* since each journal's inception. We classify research diversity based on 1) research paradigms: design and behavioral science, 2) design science research outputs, as well as 3) research methodologies. Our overall empirical evidence from the three journals shows diverse research activities in both behavioral and design science paradigms as well as in research methodologies. Even though database research is often perceived to align with design science, our study demonstrates that there exists a great extent of diversity in database research that spans both design and behavioral science among IS scholars. This diversified research agenda could provide a fertile incubator for truly trans-disciplinary scholarly work that can help set us apart from other disciplines.

This paper contributes to the design science research community in several ways. First it reviews database research published in the three premier IS journals and shows that IS scholars engage in database research from various perspectives such as system efficiency and performance, user interfaces as well as organizational capability. In addition, this paper attempts to add to the discussion of research diversity and the important place of design science research in the IS discipline. As shown in our findings, IS scholars engage in research activities from a broad spectrum of methodologies and reference disciplines. As a result, the IS community is in a unique position to take advantage of the wealth and breadth of knowledge among our colleagues through collaboration that bridges across different, but potentially synergistic, perspectives. Finally, our classification of database research according to research paradigm, methodologies, and research outputs can guide future design science scholars in identifying the most appropriate outlet for their work among the three top-tier journals reviewed in this research. With the results presented in this paper, future researchers can also find novel avenues of enquiry by examining what was previously published.

2 Database Research in the IS Discipline

The area of database research has long been a quintessential part of the body of knowledge in the IS discipline. First, the study of databases (including topics on database management and database design) has been an integral part of the information systems discipline. In fact, both the fields of database and information systems research have been in existence alongside one another since their inception in the 1960s [41, 48]. An assessment of early IS research submitted to the then-nascent *Information Systems Research* between 1987 and 1992 placed database research in two of the eight main thematic areas of IS research [44]. More recently, Vessey et al. [48] placed database research as one of the eight major topics pursued by IS scholars. In addition, most, if not all, IS researchers consider the field of data management and database management systems as part of the core knowledge of the IS discipline (see for example [10, 11, 34, 45]). The co-existence between database and IS research allows us to track the publication records of database research in IS journals since the early development of the IS community.

Second, not only has database management become an integral component of financial, accounting, and other business systems, but it has also become one of the most essential tasks performed by IS professionals [10, 20]. In fact, database management is one of the most fundamental topics taught in almost all undergraduate and graduate programs in IS [49]. In 2002, a task force of 40 prominent IS researchers put forth guidelines that include database management as one of the "key information systems concepts" that must be taught in business school curricula [23]. Database topics are also included in the IS curriculum guidelines recommended by the Association for Information Systems (AIS) at both graduate and undergraduate levels [19, 20].

The area of database research is also relatively well-defined, compared to other IS research topics. In order to distinguish database research from other IS research areas, we follow an approach similar to that by Vessey et al. [48] and Palvia et al. [37]. In particular, the classification of IS research subjects by Palvia et al. [37] classifies the study of databases and database management systems as one of the 33 main categories, distinctive from other topics such as decision support systems, knowledge management, multimedia, and systems development. Vessey et al. [48] also classified data management and databases in a separate category from other topics such as Decision Support Systems (DSS), process management, and systems management. Thus our approach to make a clear distinction between database research and other topics is in alignment with these oft-cited studies, as opposed to a more inclusive approach (e.g. [8]).

3 Database Research in Premier IS Journals

In addition to the IS discipline, Database research has also been defined as part of the computer science and engineering disciplines and "devoted to the study of the problems of managing large volumes of data" [27]. This overlapping of interests between IS and computer science/engineering researchers in the field of database offers opportunities for inter-disciplinary scholarship, but also poses a challenge among these researchers in terms of their choice of publication outlets. In addition to the main stream IS journals, such as *ISR, JMIS,* and *MISQ*, some other possible publication outlets for IS researchers include both journals that are classified as primary interest for computer scientists and journals that specialize in database management topics (e.g.. *Communications of the ACM, ACM Transactions on Database Systems (ACM TODS), IEEE Transactions on Knowledge and Data*

Engineering (IEEE TKDE), Data and Knowledge Engineering, and *Journal of Database Management (JDM)*). However, even with high-quality journals specializing in database research such as *ACM TODS* and *JDM*, the three mainstream IS journals (i.e., *ISR, JMIS,* and *MISQ*) remain the top three premier publication outlets for IS researchers in academia, especially for the purpose of tenure and promotion evaluation [15, 47]. Past studies also recognize these same three journals as among the most prestigious publication outlets in our field (see, for example, [8, 24, 39, 48]). As a consequence, we choose to include published articles from these three IS journals (*ISR, JMIS,* and *MISQ*) in our analysis. We obtain a copy of all research articles that have appeared in the IS journals from their inception to 2007. We exclude editor's notes/comments and interviews and retain only research articles. The remaining articles are then read and coded based on the classification scheme described below.

We first determine whether or not each article that appeared in the three journals is database research. To help distinguish database research from other topics, we use the subject classification specified in Palvia et al. [37]. This classification scheme is derived from extensive research published by Alavi & Carlson [1] and Barki, Rivard, & Talbot [5]. As we noted earlier, this classification framework separates research in databases and database management systems (our main focus) from other topics such as decision support systems, knowledge management, multimedia, and systems development. Table 1 presents the number of articles coded by the three researchers as database research. We went through several phases of coding and discussion to obtain the final list of database articles. We recognize that whether or not an article is database research may not be apparent from the article's title alone, therefore we take great care to consult with both the abstract and the actual text of the articles during our coding sessions.

Considering only the database articles, we then develop a classification scheme which includes the following dimensions: design versus behavioral science, design science research outputs, and research methodologies used. In order to provide broader insight into the profile of database research, we also present a cross-analysis for each of the three journals. In order to provide empirical evidence on these classifications, we use content analysis as the primary method.

Table 1. Total number of database articles by journal

Journal	ISR	JMIS	MISQ	Total
Number of database articles	19	38	19	76
Time period covered	1990-2007	1984-2007	1977-2007	1977-2007
Number of database articles per year	1.05	1.58	0.61	2.45
1990 – 2007				
Number of database articles	19	20	7	46
Number of database articles per year	1.05	1.11	0.39	2.55

4 Design Science and Behavioral Science Research

In this section, we focus on the design versus behavioral science research paradigms. In their seminal work, Hevner and colleagues [22] presented a guideline for conducting design science research in IS, adumbrating a decisive change for design science researchers within the mainstream IS community. Their work has been used by IS journals as a touchstone to help determine the validity of scholarly work for publication. This restored interest between design and behavioral science research has renewed discussion about expanding publication opportunities at some of our most prestigious journals in order to foster a more diverse research agenda (see, for example [43]).

Research in IS has been defined as dealing primarily with a complex system consisting of computer hardware, software, data, procedures, decision models and people [31]. IS research therefore focuses on "artificial" phenomena that involve tools, techniques, and materials designed and implemented by humans to achieve predefined objectives [9]. As pointed out by March & Smith [31], these artificial IS phenomena can be both "created and studied," placing the field of IS research at a crossroads between natural and design science. Yet, past studies have shown the apparent preference of mainstream IS journals for behavioral research (see, for example, [4, 28]). In recent years, many prominent IS researchers have argued for more focus on design science research and for greater recognition of design science research among mainstream IS journals (see, for example [22, 31]).

According to Hevner et al. [22], "IS research occurs at the confluence of people, organizations and technology; therefore two distinct and complementary paradigms are necessary to acquire the information required to improve information systems: (1) behavioral science and (2) design science." They describe these two paradigms as follows:

> "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."[22, p.76].
>
> "The design-science paradigm has its roots in engineering and the sciences of the artifact ([42]). 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, and use of information systems can be effectively and efficiently accomplished ([46]; [13])."[22, p. 76].

Hevner et al. [22], March & Smith [31], and Nunamaker et al. [35] further describe the definition of IT artifacts that are essential outputs of IS research. These artifacts consist of four general outputs for design science research which include *constructs, models, methods, and instantiations*. Table 2 summarizes the description of the four research outputs.

Table 2. Design Science Research Outputs

Output	Description
Constructs	The conceptual "vocabulary and symbols used to define problems and solutions" of a domain [22]. They include "linguistic devices to define and communicate problems [8]."
Models	"A set of propositions or statements expressing relationships between constructs [28]."
Methods	A set of related procedural steps used to perform a task. Specifically, these steps are used to "define solution processes through formal algorithms or step-by-step procedures [8]."
Instantiations	An implementation of constructs, models and methods in a working or prototype system. [8, 28]

Although database research could fall under either design or behavioral science paradigms, the focus of database research is often perceived to align with the design science research paradigm. In fact, scholarly articles often use topics in database to demonstrate guidelines for design science research in IS (see, for example, [22, 28, 31]). This apparent association between database and design science research inadvertently places IS researchers with an interest in database topics at odds with the coverage of mainstream IS journals. Even though database research shares its roots with computer science and engineering, in what follows, we demonstrate that there exists a great extent of diversity in database research that spans both design and behavioral science among IS scholars. This diversified research agenda, like other research areas in IS, could provide a fertile incubator for truly trans-disciplinary scholarly work that can help set us apart from other disciplines.

In order to determine whether each database article is design science or behavioral science research, we follow the guidelines proposed by Hevner et al.[22]. Then, we use the definition of IT artifact outputs as described in [31] and further defined by Hevner et al. [22], Benbunan-Fich & Mohan [8], and Kuechler et al. [28]. To further

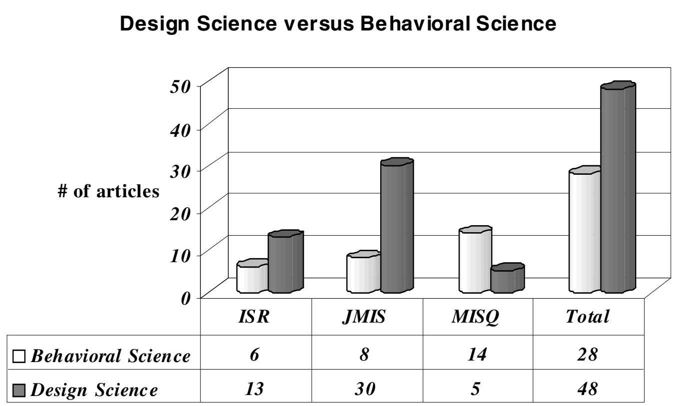
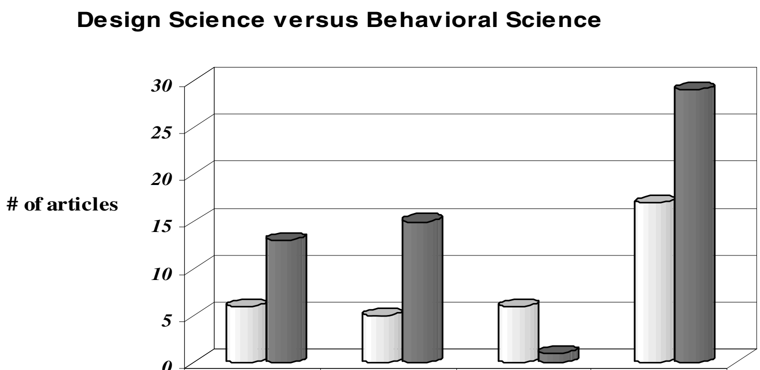


Fig. 1a. Number of Database Articles since Journal Inception to 2007



	ISR	JMIS	MISQ	Total
□ Behavioral Science	6	5	6	17
■ Design Science	13	15	1	29

Fig. 1b. Number of Database Articles from 1990 to 2007

differentiate the design science research articles, we focus on the four primary outputs of design science research: constructs (vocabulary and symbols), models (abstractions and representations), methods (algorithms and practices), and instantiations (implemented and prototype systems). Some articles may present multiple IT artifacts as their research outputs. In such situations, we focus on the article's primary research objective and identify the article with only one of the four output categories. Fig. 1a and Fig.1b present the results of our coding for each journal.

As shown in Fig.1a, out of the 76 database articles, 48 articles are classified as design science research and the remaining 28 articles are classified as behavioral science articles.

In addition, Fig.1b shows that during the time period from 1990 to 2007, 29 database articles are design science research and 17 articles are classified as behavioral science research, totaling 46 database articles during this period. Furthermore, our examination of the design science outputs since the three journals' inception (see Fig. 2a) shows that 26 articles propose *methods* as their primary design science output, followed by *models* (12 articles), *instantiation* (7 articles), and *constructs* (3 articles). Furthermore, out of the 29 design science articles published since 1990, 17 articles propose *methods* as their primary output, followed by *models* and *instantiation* (5 articles each) and *constructs* (2 articles). Table 3 also shows the breakdown for each journal. For all three journals, we find *methods* to be the most common design science research output.

Combining the results from the three journals, we show that there exists diversity in terms of the two research paradigms. However, this evidence of diversity is not consistently found in all the three journals examined. As can be seen, while database research is primarily design science for *ISR* and *JMIS*, it is primarily behavioral in the case of *MISQ*.

Table 3. Design Science Database Research Output

Design Science Output (since journal inception to 2007)	ISR	JMIS	MISQ	Total
Constructs	1	2		3
Instantiation		6	1	7
Methods	10	13	3	26
Models	2	9	1	12
Total	13	30	5	48
Design Science Output (from 1990 to 2007)				
Constructs	1	1		2
Instantiation		4	1	5
Methods	10	7		17
Models	2	3		5
Total	13	15	1	29

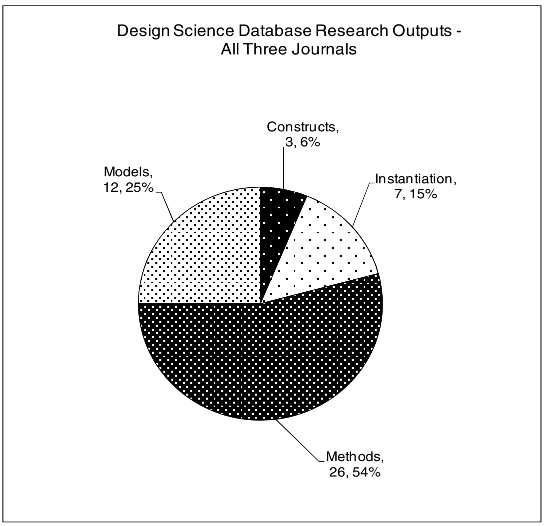


Fig. 2a. Design Science Database Research Outputs (since journal inception to 2007)

5 Research Methodologies in Database Research

We also classify the articles in terms of the research methodologies used. We include the classification on research methods in our study because research methods represent "the means for gaining knowledge" and "may be used with any epistemological perspective" [12].

Past studies (e.g., [1, 16, 48]) have identified several research methods that are commonly used in IS research. In order to capture all of the research methodologies represented in the three journals, we follow a more up-to-date classification framework proposed by Palvia et al. [37] as shown in Table 4. In addition, since it is possible that a research article may rely upon multiple methodologies, the coders

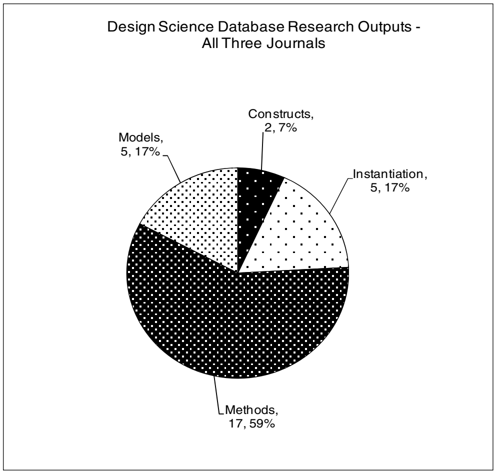


Fig. 2b. Design Science Database Research Outputs (from 1990 to 2007)

record up to two research methodologies for each article. A similar approach was employed by past studies on the epistemology of IS research [36, 37].

Tables 5 and 6 present the results of our coding for the primary research methodology, cross-tabulated with the research paradigms. As shown in the two tables, behavioral science database research primarily uses laboratory experiment and survey methods, whereas design science database research involves mathematical models as well as frameworks and conceptual models.

An examination of these results for each journal shows some interesting patterns (see Figures 3a and 3b). In *ISR*, one methodology clearly dominates each research paradigm - laboratory experiments in behavioral science research and mathematical models in design science research. Although the numbers are very small, the behavioral science research in *JMIS* has evidence of laboratory experiment and survey methodologies.

Table 4. Research Methodologies [37]

1. Speculation/commentary
2. Frameworks and Conceptual Model
3. Library Research
4. Literature Analysis
5. Case Study
6. Survey
7. Field Study
8. Field Experiment
9. Laboratory Experiment
10. Mathematical Model
11. Qualitative Research
12. Interview
13. Secondary Data
14. Content Analysis

Table 5. Research Paradigm and Primary Research Methodology (since inception to 2007)

Research Paradigm	Methodology	ISR	JMIS	MISQ	Total
Behavioral Science	Case Study			3	3
	Field Study			1	1
	Laboratory Experiment	5	2	5	12
	Library Research		1	2	3
	Literature Analysis		1	1	2
	Mathematical Model	1	1		2
	Survey		3	2	5
Behavioral Sc.Total		6	8	14	28
Design Science	Frameworks & Conceptual Model		10	5	15
	Laboratory Experiment		1		1
	Mathematical Model	13	19		32
Design Sc. Total		13	30	5	48
Total		19	38	19	76

In design science research, *JMIS* published articles that use frameworks and conceptual models, in addition to the most commonly used mathematical model methodology. The design science articles in *MISQ* all use frameworks and conceptual models as the research methodology. On the other hand, the behavioral science articles in *MISQ* include laboratory experiments, case studies, surveys, and library research. For the two research paradigms combined, mathematical models (44%) are the most commonly used research methodology in database research, followed by frameworks and conceptual models (15%) and laboratory experiments (17%).

Table 6. Research Paradigm and Primary Research Methodology (from 1990 to 2007)

Research Paradigm	Methodology	ISR	JMIS	MISQ	Total
Behavioral Science	Case Study			1	1
	Field Study				
	Laboratory Experiment	5	2	3	10
	Library Research				
	Literature Analysis			1	1
	Mathematical Model	1	1		2
	Survey		2	1	3
Behavioral Science Total		6	5	6	17
Design Science	Frameworks & Conceptual Model		3	1	4
	Laboratory Experiment		1		1
	Mathematical Model	13	11		24
Design Science Total		13	15	1	29
Total		19	20	7	46

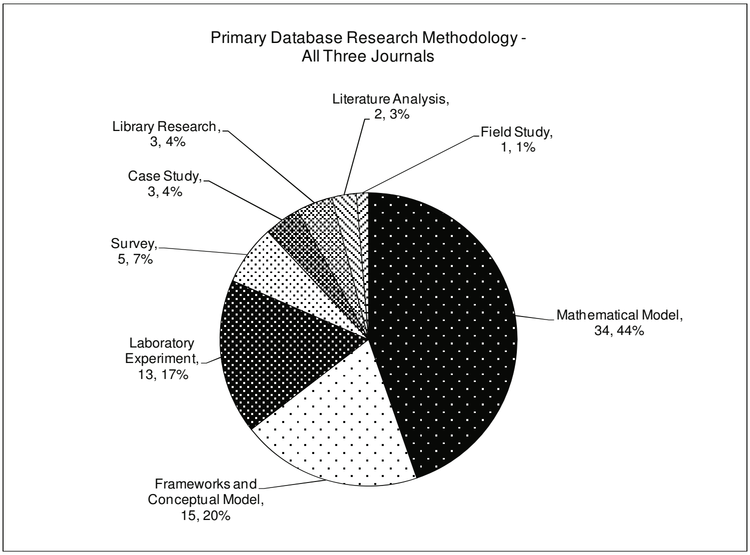


Fig. 3a. Primary Database Research Methodology (since inception to 2007)

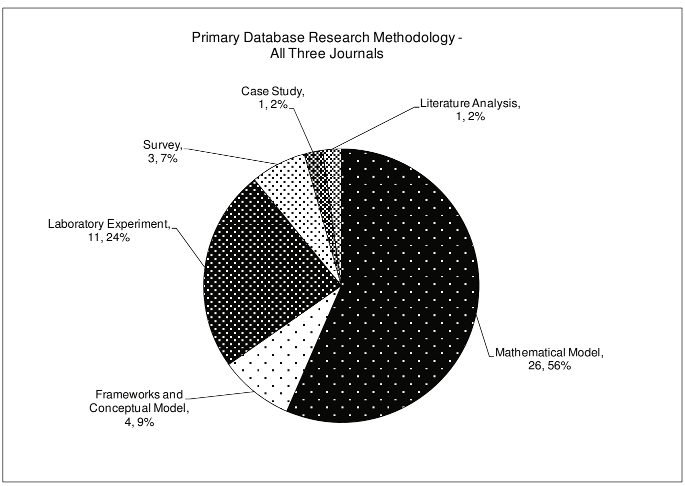


Fig. 3b. Primary Database Research Methodology (from 1990 to 2007)

6 Conclusion

In this article, we profile database research published in the top three premier IS journals. Our empirical evidence shows diversity in research methodology, IT construct, and research paradigm even within a specific, supposedly well-defined topic such as database management. In addition to showing the variety of database research published in the three journals as a whole, this paper also provides some insights regarding the focus of each journal in terms of research paradigm and methodology.

This paper contributes to the design science research community in several ways. First it reviews database research published in the three premier IS journals (i.e., ISR, JMIS, and MISQ). Our empirical evidence shows diversity in research methodology, IT construct, and research -paradigm (i.e., design and behavioral science) even within a specific topic such as database management. Our result shows that IS scholars engage in database research from various perspectives such as system efficiency and performance, user interfaces as well as organizational capability. Future research can extend the current results into other IS journals and topics in order to provide more comprehensive insights into design science research. We also hope that our paper will add to the discussion of research diversity in IS and the important place of design science.

Finally, our classification of database research according to research paradigm, methodologies, and research outputs can guide future design science scholars in identifying the most appropriate outlets for their work. With the results presented in this paper, future researchers can also find novel avenues of enquiry by examining what was previously published.

This study has some limitations. The study focused on database management research and included only articles published in three premier journals until the year 2007. Our future research will extend the scope of this study and examine if the results hold for recent years and other premier journals and topics in the IS field.

References

1. Alavi, M., Carlson, P.: A Review of MIS Research and Disciplinary Development. J. Manage. Inform. Syst. 8(4), 45–62 (1992)
2. Applegate, L.M., King, J.L.: Rigor and Relevance: Careers on the Line. MIS Quart. 23(1), 17–18 (1999)
3. Ayanso, A., Lertwachara, K., Vachon, F.: Diversity or Identity Crisis? An Examination of Leading IS Journals. Commun. of the AIS 20, 660–680 (2007)
4. Bajaj, A., Batra, D., Hevner, A., Parsons, J., Siau, K.: Systems Analysis and Design: Should We be Researching What we Teach? Commun. of the AIS 15, 478–493 (2005)
5. Barki, H., Rivard, S., Talbot, J.: A Keyword Classification Scheme for IS Research Literature: An Update. MIS Quart. 17(2), 209–226 (1993)
6. Benbasat, I., Weber, R.: Research Commentary: Rethinking ‘Diversity in Information Systems Research. Inform. Syst. Res. 7(4), 389–399 (1996)
7. Benbasat, I., Zmud, R.W.: The Identity Crisis within the IS Discipline: Defining and Communicating the Discipline’s Core Properties. MIS Quart. 27(2), 183–194 (2003)
8. Benbunan-Fich, R., Mohan, K.: Research publications in Systems Development during 2000-2004. Commun. of the AIS 17, 373–390 (2006)
9. Davis, G.B.: Management Information Systems: Conceptual Foundations, Structure, and Development. McGraw-Hill, New York (1974)
10. Davis, G.B.: The Knowledge and Skill Requirement for the Doctorate in MIS. In: Proceedings of the First International Conference on Information Systems (ICIS), Philadelphia, PA, December 8-10, pp. 174–186 (1980)

11. Davis, G.B.: Information Systems as an Academic Discipline: Looking Back, Looking Forward, and Ensuring the Future. In: Proceedings of the IFIP International Federation for Information Processing: The Past and Future of Information Systems: 1976-2006 and Beyond, vol. 214, pp. 11–25 (2006)
12. DeLuca, D., Kock, N.: Publishing Information Systems Action Research for a Positivist Audience. Commun. of the AIS 19, 183–204 (2007)
13. Denning, P.J.: A New Social Contract for Research. Communications of the ACM 40(2), 132–134 (1997)
14. Dennis, A.R.: Relevance in Inform. Syst. Res. Commun. of the AIS 6, 1–6 (2001)
15. Dennis, A.R., Valacich, J.S., Fuller, M.A., Schneider, C.: Research Standards for Promotion and Tenure in Information Systems. MIS Quart. 30(1), 1–12 (2006)
16. Farhoomand, A.F., Drury, D.H.: A Historiographical Examination of Information Systems. Commun. of the AIS 1, 1–20 (1999)
17. Galliers, R.D.: Change as Crisis or Growth? Toward a Trans-disciplinary View of Information Systems as a Field of Study: A Response to Benbasat and Zmud's Call for Returning to the IT Artifact. Journal of the Association for Information Systems 4, 337–351 (2003)
18. Galliers, R.D., Meadows, M.: A Discipline Divided: Globalization and Parochialism in Inform. Syst. Res. Commun. of the AIS 11, 108–117 (2003)
19. Gorgone, J.T., Davis, G.B., Valacich, J.S., Topi, H., Feinstein, D.L., Longenecker, H.E.: IS 2002 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems. Commun. of the AIS 11, 1–48 (2003)
20. Gorgone, J.T., Gray, P., Stohr, E.A., Valacich, J.S., Wigand, R.T.: MSIS 2006: Model Curriculum and Guidelines for Graduate Degree Programs in Information Systems. Commun. of the AIS 17(1), 121–195 (2006)
21. Guthrie, R.A.: The IS Core- V Defining the IS Core. Commun. of the AIS 12, 557–561 (2003)
22. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quart. 28(1), 75–106 (2004)
23. Ives, B., et al.: What Every Business Student Needs to Know about Information Systems. Commun. of the AIS 9, 467–477 (2002)
24. Katerattanakul, P., Razi, M.A., Han, B.T., Kam, H.: Consistency and Concern on IS Journal Rankings. J. of Inform. Technol. Theory and Appl. 7(2), 1–20 (2005)
25. Keen, P.G.W.: MIS Research: Reference Disciplines and a Cumulative Tradition. In: McLean, E.R. (ed.) Proceedings of the First International Conference on Information Systems, pp. 9–18 (1980)
26. King, J.L.: Editorial Notes. Inform. Syst. Res. 4(4), 291–298 (1993)
27. Korth, H.F., Silberschatz, A.: Database Research Faces the Information Explosion. Commun. of the ACM 40(2), 139–142 (1997)
28. Kuechler, B., Vaishnavi, V., Kuechler, W.L.: Design [Science] Research in IS: A Work in Progress, Association for Information Systems (2008), http://ais.affiniscape.com/displaycommon.cfm?an=1&subarticlenbr=386#ISDRhistory
29. Landis, J., Koch, G.: The Measurement of Observer Agreement for Categorical Data. Biometrics 33(1), 159–174 (1977)
30. Lee, A.: Editor's Comments. MIS Quart. 25(4), v–xv (2001)
31. March, S.T., Smith, G.F.: Design and Natural Science Research on Information Technology. Decis. Support Syst. 15, 251–266 (1995)
32. Myers, M.D.: Qualitative Research in Information Systems. MIS Quart. 21(2), 241–242 (1997)

33. Neuendorf, K.: The Content Analysis Guidebook. Sage Publications, Thousand Oaks (2002)
34. Nolan, R.L., Wetherbe, J.C.: Toward a Comprehensive Framework for MIS Research. MIS Quart. 4(2), 1–19 (1980)
35. Nunamaker, J., Chen, M., Purdin, T.D.M.: Systems Development in Inform. Syst. Res. J. Manage. Inform. Syst. 7(3), 89–106 (1991)
36. Palvia, P., Mao, E., Salam, A.F., Soliman, K.S.: Management Inform. Syst. Res.: What's There in a Methodology? Commun. of the AIS 11, 289–309 (2003)
37. Palvia, P., Mao, E., Midha, V., Pinjani, P., Salam, A.F.: Research Methodologies in MIS: An Update. Commun. of the AIS 14, 526–542 (2004)
38. Peffers, K.: Editorial: IS Research Using Theory from Elsewhere. J. of Inform. Technol. Theory and Appl. 9(2), iii–vi (2008)
39. Peffers, K., Ya, T.: Identifying and Evaluating the Universe of Outlets for Inform. Syst. Res.: Ranking the Journals. J. of Inform. Technol. Theory and Appl. 5(1), 63–84 (2003)
40. Robey, D.: Research Commentary: Diversity in Inform. Syst. Res.: Threat, Promise, and Responsibility. Inform. Syst. Res. 7(4), 400–408 (1996)
41. Silberschatz, A., Stonebraker, M., Ullman, J.: Database Research: Achievements and Opportunities. In: The 'Lagunita' Report of the NSF Invitational Workshop on the Future of Database Research Held in (February 1990); SIGMOD RECORD 19(4), 6–22 (1990)
42. Simon, H.A.: The Sciences of the Artificial, 3rd edn. MIT Press, Cambridge (1996)
43. Straub, D.: Editor's Comments. MIS Quart. 32(1), iii–viii (2008)
44. Swanson, E.B., Ramiller, N.C.: Information Systems Research Thematics: Submission to a New Journal, 1987-1992. Inform. Syst. Res. 4(4), 299–330 (1993)
45. Teng, J.T.C., Galletta, D.F.: MIS Research Directions: A Survey of Researchers' Views. Data Base 22(1/2), 53–62 (1991)
46. Tsichritzis, D.: The Dynamics of Innovation. In: Beyond Calculation: The Next Fifty Years of Computing, New York, NY, Copernicus, pp. 259–265 (1997)
47. Valacich, J.S., Fuller, M.A., Schneider, C., Dennis, A.R.: Publication Opportunities in Premier Business Outlets: How Level is the Playing Field? Inform. Syst. Res. 17(2), 107–125 (2006)
48. Vessey, I., Ramesh, V., Glass, R.: Research in Information Systems: An Empirical Study of Diversity in the Discipline and Its Journals. J. Manage. Inform. Syst. 19(2), 129–174 (2002)
49. Vician, C., Garfield, M., Hoffer, J.A., Prescott, M., Rollier, B., Strong, D.M., Elder, K.L.: The AMCIS 2003 panels on IS Education-II the Chicken and the Egg Debate: Positioning Database Content in the Information Systems Curriculum. Commun. of the AIS 14, 147–157 (2004)

Appendix A: Reliability Measure in Content Analysis

For each journal, the three authors worked independently to code the articles, following the content analysis procedure outlined in [33]. The total numbers of reviewed articles since the inception of *ISR*, *JMIS*, and *MISQ* were 362, 809, and 805, respectively. The numbers of reviewed articles for *ISR*, *JMIS*, and *MISQ* since 1990 were 362, 659, and 472, respectively. We conducted content analysis to classify the 76 database research articles on the research methodologies, design versus behavioral science classifications, and the design science research outputs. Before the classification process started, we documented the definitions of each classification categories as described above. The coders then discussed and came to an initial agreement on the interpretation of these definitions. The percentage of agreement and Cohen's kappa were used as measurement of inter-coder reliability.

Typically, kappa values between 0.61 and 0.80 are regarded as "substantial," and those greater than 0.80 are deemed "almost perfect" [29]. Table 7 presents the results of the inter-coder reliability analyses.

Table 7. Inter-rater Reliability Measures

Classification frameworks	*Percent of Overall Agreement*	*Kappa Statistics*
Design versus Behavioral Science	0.9474	0.8947
Design Science Outputs	0.8841	0.8454
Research Methodologies	0.9386	0.9339

Design Science Research Demonstrators for Punctuation – The Establishment of a Service Ecosystem

Daniel Rudmark[1,2] and Mikael Lind[2,1]

[1] University of Borås, School of Business and Informatics,
501 90 Borås, Sweden
[2] Viktoria Institute, Hörselgången 4,
417 56 Göteborg, Sweden
daniel.rudmark@hb.se
mikael.lind@viktoria.se

Abstract. Design Science Research (DSR) is concerned with demonstrating design principles. In order to prove the utility of these principles, design ideas are materialized into artifacts and put into an environment sufficient to host the testing of these principles. When DSR is used in combination with action research, constraints in the environment may restrain researchers to fully inscribe or test such principles. In this paper it is argued that scholars pursuing DSR has paid insufficient attention to the type of change necessary in the local practice. We draw upon theories on IS change as punctuated equilibrium to illustrate when DSR demonstrators can be used to make substantial contributions to local practice as well as to the scientific body of knowledge.

Keywords: Demonstrator, Action Research, Design Science Research, Punctuated Equilibrium

1 Introduction

In the information systems research community there has been a prevailing debate on how the discipline can maintain relevance for the practice it sets out to study [1]. It has been argued that IS as a discipline has paid insufficient attention on developing knowledge about IT artifacts as such [2] and that design science research (DSR) lends a promise of addressing this dearth [3], and by doing so, offering a potential to increase practitioner interest.

Within DSR, scholars has theorized on how to conceptualize scientific knowledge about artificials and disseminate it among researchers as well as practitioners [4] [5] [6]. In fact, within DSR there has been a comparably strong focus on how to make a difference for practice. As for example, one of the heavily cited DSR guidelines by Hevner and associates [4] stipulates that the significance of the research pursued should partially be judged by its findings' having been properly communicated to practitioners. Moreover, several authors have discussed how certain types of more socio-technical DSR conducted in clinical, real-world settings may yield more relevant findings (than those performed in lab-like environments). Although disputed [7],

H. Jain, A.P. Sinha, and P. Vitharana (Eds.): DESRIST 2011, LNCS 6629, pp. 153–165, 2011.

this mode of DSR research is argued to generate design knowledge of interest for the local practice as well as the scientific community and general practice [8]. An unresolved quest is however how practitioners can and should be part of research-informed development.

A central concept within DSR is the IT artifact. The role of the IT artifact in the research process is manifold; it has been described as an inquiring instrument [9], as the principal outcome of the research endeavor [4], and/or as a validation of the generated design knowledge [5]. Further, if the researchers engage in more action-oriented DSR the IT artifact also constitutes the basis for an intervention solving a local organizational problem (c.f. Sein and associates [6]). Whatever the purpose, the materialized artifact must encompass the developed design knowledge (i.e. design principles). For researchers striving to pursue in-situ DSR (in development, evaluation or both), this fact poses some serious challenges. Since DSR is concerned with novel, insofar unsolved organizational problems [10], the artifacts embodying the design knowledge often take the form of functional prototypes, ideally effectively demonstrating the idea pursued. As a consequence, these *demonstrators* [11] may i.e. lack sufficient integration with existing IS architecture [12] than more production-like systems would, or, in the case of altering an existing system, the inscription of the design knowledge may be partially unfeasible [13]. Either way, user utility in the local practice (an important research evaluation criterion) is hampered. Considering this background there seems to be insufficient knowledge of when a DSR demonstrator may be of particular benefit for both researchers and practitioners. Thus, in this paper we will explore how the concept of a demonstrator may be able to combine researcher requirements and practitioner relevance through the following research question:

- *how and when can DSR demonstrators contribute to sustainable change of practice while maintaining scientific contributions?*

The rest of the paper is organized as follows: In the next section a theoretical account of design science research as research approach and intervention is given. This is followed by a theoretical model of IS change before an analysis of the empirical material is presented. Lastly we provide a discussion before returning to the research question in the conclusions.

2 Related Literature

In recent years, framing systems development as design has rendered great interest within the community of information systems research. Although different streams of such research exist [14], the by far most cited design research approach is Design Science Research (DSR) [14] [15]. In DSR, knowledge about and understanding of a problem domain and its corresponding artifact-based solutions are achieved through a scientifically grounded implementation and evaluation of artifacts [4]. Hence, in DSR the researcher-as-designer is stressed. The design knowledge developed may either concern an *improvement* of an existing entity or *construction* knowledge on how to design a new unprecedented artificial [16].

In recent years, DSR and DSR-like approaches have also been used as means in theory-informed intervention (c.f. [13] [17] [18] [19]). Our literature review suggests that a demarcation line seems to exist for when such intervention is beneficial for in DSR. Some scholars (c.f. Venable [20]) argue that proper theorizing requires that initial hypothesis generation (and its materialization through an artifact) is best performed outside the organizational context. Thus, the core idea developed avoids being too ingrained with the situated problems at issue. After the artifact is developed one option for evaluation (among many) is action research where the artifact is evaluated based on the performance in an authentic setting. In a recent theorizing effort an alternative approach is presented by Sein et al. [6]. They consider the DSR artifact to correspond the ensemble concept [2]. As a consequence, the expected and unforeseen consequences of introducing an artifact into an authentic environment constitute the basis for generalized learning. Thus, they suggest a more reciprocal process where the researcher and the client system are rather engaged in ongoing iterations between development and evaluation.

Whatever the approach to intervening through a DSR IT artifact is, it seems that this type of clinical hypothesis testing is associated with considerable challenges. Since the incentives to develop such technology are novelty and invention [10] [20], it is not surprising that research in this vein typically suffer from incongruence between artifact requirements and realities in the organizational environment. As a consequence, the functionality of the artifact risks not being fully utilized and thereby hampering the dual interests of action research [21]. The environment subjected to change does not exhibit a “clean slate” but carries a socio-technical trajectory (which is to be altered into a more preferred state). Therefore it is not surprising that interventional DSR demonstrators suffers from i.e. inability to integrate with existing architectures [12], insufficient input data sets [19], mutability issues [13] or inability to cover the necessary variety of hardware configurations [18]. On the other hand, it has also been argued that it is ethically questionable to expose a research client to the risks that come with using novel technologies in an organizational context [8]. In sum, successful DSR intervention requires not only that the boundaries of the research interests and the organizational coincide [21] but also that the practitioner system allows for necessary alteration as well as a mutual understanding of what an engagement in action research implies [22].

Given these challenges in developing and/or evaluating DSR IT artifacts through intervention, we believe that IS DSR as a field needs a more precise understanding under what conditions such theory-informed intervention stands a greater chance of serving the dual agendas of action research [21]. While IS as a field has a long history of examining the unfolding of IT-related change processes, this understanding remains underdeveloped within IS DSR. In what follows, we thus present a theoretical lens on IS change as described by Lyytinen and Newman [23].

3 Theoretical Basis – IS Change as Punctuated Equilibrium

It has recently been argued that IS change often follow a punctuated model [23] in which a multi-level socio-technical perspective of IS change is employed. This line of reasoning carries four main units of analysis: work system, building system, organizational context and environmental context. The *work system* is concerned with the *de facto* IS operations and is typically deeply embeddedin daily IS operations. It is further characterized by its "low malleability due to path dependencies, habitualization, cognitive inertia, and high complexity" ([23] p. 592). One example of such a system is the organization of IS departments where i.e. roles related to used methodologies (e.g. project managers, developers etc.) can be found. A *building system* is typically locally configured to address a specific problem. In analogy with the previous example, a building system could be compared with a specific project configuration. The work and build system are embedded in the *organizational context* which in part is residing in an *environmental context*.

A need for change is built up when a gap between one or more core organizational components; (task, structure, people and technology [24]) within the work system surfaces. The gap is preceded by a *critical incident*, stemming from within the organization or the outside environment. One example would be i.e. if an established systems development method within the work system fails in adequately addressing organizational problems. In such a case a new building system (typically positioned somewhat independent of the work system) is set up to intervene in an attempt to bridge the gap(s). In successful cases, the building system intervention may eliminate the dissonance between the core organizational components and the work system *equilibrium* may continue to reign. However, in other cases the changes necessary to address the gap may require the existing work system to be subverted - *punctuation* is needed. Punctuations "add novel technical elements, replace, remove or expand organizational structures and routines, and wipe out ideas, beliefs, skills, and values that underpin and are embodied in the organization" ([23], p. 594). Thus in order to resolve such "wicked" gaps, changes in the deep structures of the work system is necessary to occur before a new equilibrium may begin. Using this theoretical backdrop we now examine the build-up and early phases of a DSR project (which the authors of this article are actively taking part in) where such type of change is diagnosed as necessary and its implications for DSR demonstrators.

4 Method

4.1 Research Site

The nexus of the data collection activities has been a working group concerned with interorganizational cooperation concerning information technology between a Swedish region's public traffic authorities. The main task of this working group is to coordinate traffic-related information system development activities spanning these three organizations. The working group became engaged in a larger research setting, a research program focusing on how to innovate more sustainable everyday travel through the use of information technology. This research program (encompassing cross-disciplinary research) explore two inter-linked research questions; *how to enable infrastructure innovation?*and*how to enable distributed service innovation?* These larger research questions are explored by the support of DSR-enabled demonstrators. This means that the setting facilitated by the research program become suitable for studying how DSR demonstrators can contribute to sustainable change of practice while maintaining scientific contributions.

The setting included a variety of research partners undertaking different roles; i.e. information providers, brokers, mediators, developers, consumers of data, and consumers of services in a service ecosystem [25]. Even though these roles would potentially challenge existing deep structures within the participating organizations, due the situation described below, it was yet conceived a necessity by these organizations to explore these roles within a larger eco-system. The research program has therefore an ambition to enable an open innovation eco-system making it possible for distributed development of services based on data from diverse sources. The major challenge facing these organizations concerned how to design the infrastructure in such a way that development of end-user applications supporting sustainable every day travel was encouraged. The events that this paper builds upon occurred between February 2008 and June 2010.

The research approach undertaken to develop the knowledge in this paper builds upon an interplay between collecting experiences from the DSR setting (the research program), categorization of these experiences in relation to both design science theories as well as theories on IS change for punctuation. The identification of the empirical experiences is driven from the research question explored in this paper. By this theoretical and empirical interplay the use of DSR demonstrators in punctuation has been identified (see section 6).

4.2 Data Collection

The data presented in this paper has been collected over 12 months (2009-2010) as part of diagnosing phase in a canonical action research project [22] in progress. The data consisted of working group meetings, interviews, internal reports, publically available reports and field notes. All analyzed meetings and interviews have been audio taped and transcribed. Since a substantial part of the informants were engaged in the action research project it gave good access to data (such as recordings of meetings). The data were collected in three phases. First, in order to better understand the system development challenges the group was facing, the first set of data collected was interviews of explorative nature alongside with working group meetings (to better understand the current challenges). Second, the next set of interviews with these actors focused on previous projects and the unfolding of events. In this step internal and publically available reports mentioned in previous data were also collected. Thirdly, interviews with outside actors such as previous employees and outside developers were conducted. In total 13 interviews were conducted with 9 respondents, 4 monthly working group meetings were attended and 6 reports were collected. All in all, this resulted in an analysis of a few hundred pages of written material.

4.3 Data Analysis

All transcribed audio material and reports has been analyzed using atlas.ti, a data analysis software package. Utterances and report paragraphs addressing the scope of research have been identified and coded accordingly. In phase 1 (as described above) data was analyzed inductively line-by-line inspired by the methods of Strauss and

Corbin [26]. In this stage it was important to diagnose the current state of affairs in the local practice without forcing too much researcher preconceptions onto the data while still maintaining scientific integrity [27]. The relationships between codes were established and a detailed snapshot of the current struggles of the working group with respect to system development approaches emerged.

As the current problematic situation became more articulated, there was a need understand the historical processes and events which led to the current gaps. To this end the data from interview round two and a number of reports were coded using the theoretical raster of punctuated IS change [23] as described above. Finally, the reconstruction of historical events was triangulated by interviews with former employees and outside developers.

5 Findings

5.1 A Gap Arising

The transfer and processing of traffic information from data gathering points out to travelers' has been the backbone of the working group's mission ever since its inception.

"And if you say the vision for, that is the overall vision [for this group], it is getting more people to go by public transportation rather than taking the car or at least leave the car at home and get to their destination in another way. That is sort of the driving force for our entire work [in this working group]." Project Manager within a Personal Transport Authority.

In the 1990's and early 2000's the infrastructures used to spread this information (such as of display signs located at bus stops and train stations) was typically owned and controlled by these organizations. During the 2000's as a result of the massive penetration of the World Wide Web in average households, travel planning and travel update capabilities were extended into the homes and work places of most travellers.

During the autumn of 2008 there was a sudden increase in the usage of one of the public transportation company's servers. In 2002 the company set up a server hosting an xml-based web service where outside developers were given access to a selection of the company's data (such as travel planning capabilities). The existence of the server was not officially announced but was by e.g. students and smaller companies who wanted to test an idea related to public transport application development. The server address along with necessary credentials for access to the data had become available on various web forums and it was this server, which now indicated a sudden increase in usage. The reason for this increase was the release of the first (and unsanctioned) iPhone application whose introduction marked a definite breakthrough for the use of mobile technologies in the history of the public transportation company. Up to this point mobile phones had been causing a slim 1-1,5% of the total customer-initiated travel planning web traffic (the remainder was originating from the company homepage) whereas currently iPhone applications alone is producing 12,5 % of the total amounts of hits

Although positive from an information diffusion perspective, this situation faced the working group with a dilemma: to be able to draw upon this new consumer technology new and device-specific implementations were required. Moreover, just one year after the release of the iPhone application Google released their mobile platform Android which was expected to work up a significant user base as well. The common denominator between these devices was a need for platform-centric implementation, which contrasted the previous standardized html-based and hence platform-independent capabilities for mobile phones. The question thus arose on how to develop information systems inclusive of all such emerging devices:

"Because we cannot support every new mobile phone that enters the market, or all TV sets and you name it that is used to spread information. That is not our job, that is not what we're good at, and there are others that are much better than us at it. " Business Developer within a Personal Transport Authority.

5.2 Closing the Gap – Towards a Service Ecosystem

The introduction of the iPhone seemed to indicate that there existed a body of available and somewhat anonymous developers ready to build applications on platforms and devices not presently supported or even planned to be supported by these organizations. The key was thought to lie in smart exposure of data:

"[I]n some way we must offer this data, if we are a little clever, we must offer this data in a way that is really useful for us" Business Developer within a Personal Transport Authority.

The idea that by exposing data through Application Programmers' Interfaces (API's) create a service ecosystem in which a significant body of outside developers (both commercially oriented and more altruistic ones) who would create a much thicker fabric of applications supporting sustainable everyday travel, then these organizations were capable of themselves.. This ecosystem was thought to enable as e.g. suppliers of GPS navigator with data who then through incorporation of real-time traffic disturbances into their product, would enable an informed choice for travelers for a potential use of other transportation means. Hence, i.e. designing efficient API's that could cater for both developer and traveler requirements (both protocol-wise and semantically) and enabling communication with developers, were challenges which laid ahead in order to employ this model.

However as tempting as this may seem, exposure of the data necessary for realizing such applications remained a politically sensitive issue, challenging some core assumptions in the IS operations of these public organizations,: The reason for this being controversial varied among the organizations. I.e. one of the organizations had a firm policy of how to distribute data to external parties. However, the diagnosis revealed socio-technical design of the current data provision mechanism did not sufficiently support the actors in the ecosystem and the envisioned API structure was in many ways irreconcilable with the current policy (established on a national basis). For another participating organization, just the idea of having outside developers having access to their data was highly controversial:

”We have now come to that level that we have been acting in a steering committee and almost no one in this committee group realize that we will expose this data to others. When this become known some people will go nuts!” Project Manager within a Personal Transport Authority.

Especially, the issue of having someone other than these organizations construct the end-user application was conceived as problematic:

”[W]e [ourselves] can develop a solution that presents [the comparisons] in a way that may benefit public transport […] but if we release data in a way so that someone else will construct a service, we don’t have any means of control over it, I mean they might want to build a service showing how bad public transport is[…]” Business Responsible within a Public Transportation Company.

Yet, there as strong belief that when such API’s were released - even if the first version was quite crude function-wise (as to be expected from a DSR demonstrator) – it would be enough prove the hypothesis right and assist in moving these organizations towards a service ecosystem:

“[T]here must exist a simple solution to get people to get started. Because once you start develop using [data from] accidents and road conditions and traffic speed and travel times then new questions start to come “well, alright, great, this is an accident but what has happened, will it stay like this for long”, then you want more information and you want to expand your service…” Business Developer within a Personal Transport Authority.

5.3 Engaging in a Research Program – Establishing a Building System

During spring 2010 a research program was funded by multiple research funding agencies and cooperating partners was engaged. The project (in progress since 2010-04) runs over three years and have dedicated a considerable amount of funding concerned the development of DSR artifacts in the form of a Developer’s Zone and API’s.

The three authorities described in this paper were joining this project based on the following arguments: First, it gave access to an arena where they had an opportunity to interact with actors within the service ecosystem. Second, it provided a chance to develop knowledge about this new form of systems development model:

“[I]f we are interested in building knowledge about this [DeveloperZone] […] and since we have help from [removed for review] here – then it’s going to take longer before we could start testing the service ourselves, maybe, but by being engaged in this we also develop the knowledge ourselves” Project Manager within a Personal Transport Authority.

Thirdly, developing such a Developer Zone within a research program and not within the organizations was considered beneficial. One expression of this preference is that they preferred developing it separately, outside existing inter-organizational cooperation. This was due to 1) it did not fit with any existing cooperation 2) as mentioned above, it challenged existing organizational structures. In short, it seemed as if the research program, and its output in the form of DSR demonstrators, was able to offer capabilities well suited for the task at hand, to make use of an alternative API-based distributed development model to support sustainable everyday travel.

6 Discussion

In the outset of this paper, we stated that our objective has been to develop a greater understanding of how and when DSR demonstrators can contribute to sustainable local change while maintaining the possibility to make scientific contributions. In what follows we discuss how, in particular, IS change causing punctuation have to offer to interventional DSR.

Following the IS change model suggested in Lyytinen and Newman [23], a need for IS change emerges as discrepancies between core organizational components. The cause of this gap is typically connected to an event (critical incident) within or outside the organizational boundaries. In the case described in this paper, the critical incident can be traced to the introduction of iPhone. This change sparked an opportunity for propagating traveler information in new ways, yet profoundly challenged existing, deeply rooted in ISD and information sharing work systems.

- First, along with the emergence of these nomadic hardware/software bundles, attaining sufficient coverage for travellers became troublesome.
- Second, not only did the organizations lack knowledge on how to develop such end-user applications, but more importantly, the organizations perceived such task unfeasible to handle themselves over time (considering the anticipated increase of devices and platforms consuming such information).

Fortunately, the unsanctioned iPhone travel planner indicated that resources matching this demand for development were available and willing to assist in constructing end-user applications. The first aspect, attaining knowledge on i.e. a particular smart phone development could likely be resolved through an incremental change. However, the second aspect, achieving a sustainable coverage of a multitude platforms (some of which yet unknown) required altered deep structures, through the establishment of a service ecosystem.

Two central artifacts in realizing such change were 1) the developers' zone and 2) API's exposing relevant information. These artifacts were currently mostly non-existent and there was a dearth of construction knowledge [16] within the organizations. Primarily, the challenges facing the organizations seemed to orbit around control issues [28], i.e. what measures could be taken to ensure that developers constructed applications which maintained quality and credibility while adhering to the goals of these organizations (enabling sustainable everyday travel). While existing ISD approaches within these organizations implemented mature control procedure (i.e. iterative development and thorough testing of end-user applications), it seemed somewhat of a paradox to try to control this type of open ecosystem. It became even more cumbersome as control mechanisms typically implemented by proprietary software platform owners (such as input control [29]), were unavailable due to the public nature of the information and its accompanying legislation. These types of control issues have also been identified as theoretically underexplored by recent studies [29] [30].

Clearly, there was overlap in the required design knowledge development between practitioners and researchers. As a result, the research program was founded and set up with the specific goal of implementing the artifacts necessary to enable such an ecosystem. The project was in many ways operating outside the existing work systems (although fully sanctioned from management by i.e. funding the project), involving organizational representatives sharing the interest and vision of the ecosystem. Hence, we see that the demonstrators served the following explicit purposes for practice (figure 1):

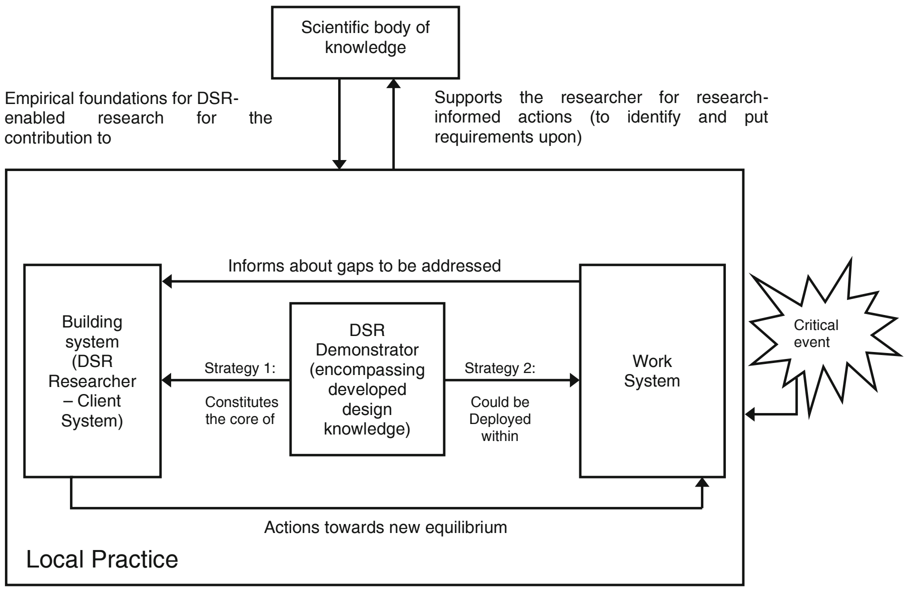


Fig. 1. The use of DSR demonstrators in punctuation

- By putting these demonstrators into use, in what can be regarded a building system [23], the hope was to demonstrate that the gap (which the introduction of the iPhone created) could be eliminated. However, since this hypothesis-testing is performed within an existing practice (and not in a lab) an altered environmental context will follow [9]. Thus, in the case that the intervention will prove successful and render useful end-user applications, the ecosystem itself would constitute a viable argument to continue on this route.
- Further, in the case that the developed technology itself used proved solid enough, the artifacts as such could constitute the architectural foundations on which the ecosystem could reside.

We argue that the reason that the demonstrators in this case carry the possibility to make a sustainable change is the type of gap that these organizations face. The establishment of a service ecosystem implies that new technology, currently absent, are implemented and the knowledge to construct it must be obtained.

Any undertaken action research endeavor must ensure the overlap between the research and practice problem [21]. However, our study suggests that organizational problems causing disrupted equilibriums [23] offer much less path-dependent legacy systems and subsequent restraining of artifact inscription and testing of the working hypotheses. Moreover, the service ecosystem challenged deeply embedded perceptions on how to conduct ISD. Thus, using the perspective of Lyytinen & Newman [23], it can be argued that a separate (and somewhat independent) building system must be set up to address the issue. We find that DSR project could be used in this role.

Thus, we argue, the theory-informed socio-technical reconfigurations undertaken by the client-researcher system stands a greater chance of succeeding, given these increased degrees of freedom. The question of *when* a demonstrator intervention stands a greater chance of succeeding, we argue, has prior to this research been missing within the DSR literature.

7 Conclusions

In summary, this paper makes three contributions. First, the research identifies that currently the DSR literature has not sufficiently addressed the *type of change* in the local practice and how this affects the suitability for sustainable impact from the DSR demonstrator. Second, by acknowledging that some IS-related organizational problems requires *punctuation*, such clinical settings offer a lesser degree of path-dependence or other organizational constraints, thus leaving more room for materializing hypothesis-testing artifacts. Third, within a DSR project addressing a punctuating change, we argue that the DSR demonstrator may be used as 1) a new, temporary practice (building system) testing the feasibility of the ideas pursued 2) serve as foundation for a new emerging work system.

The major limitation of this research is that the DSR project described in this paper is not yet completed. Thus, although a disrupted equilibrium has been identified, there remains to collect empirical evidence as to whether the shortcomings typically displayed in interventional DSR [13] [17] [18] [19] can be avoided or to a lesser degree restraining the research efforts.

References

1. Benbasat, I., Zmud, R.: Empirical Research in Information Systems: The Practice of Relevance. MIS Quarterly 23(1), 3–16 (1999)
2. Orlikowski, W., Iacono, S.: Research Commentary: Desperately Seeking the 'IT' in IT Research: A Call to Theorizing the IT Artifact. Information Systems Research 12(2), 121–134 (2001)
3. Au, Y.: Design Science I: The Role of Design Science in Electronic Commerce Research. Communications of the Association for Information Systems 7 (2001)
4. Hevner, A., March, S., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28(1), 75–105 (2004)
5. Jones, D., Gregor, S.: The Anatomy of a Design Theory (2007)

6. Sein, M., Henfridsson, O., Purao, S., Rossi, M., Lindgren, R.: Action Design Research. MIS Quarterly 35(2) (2011)
7. Iivari, J., Venable, J.: Action Research and Design Science Research - Seemingly Similar but Decisively Dissimilar. In: ECIS 2009, Verona (2009)
8. Goldkuhl, G., Lind, M.: A Multi-Grounded Design Research Process. In: Winter, R., Zhao, J.L., Aier, S. (eds.) DESRIST 2010. LNCS, vol. 6105, pp. 45–60. Springer, Heidelberg (2010)
9. Purao, S.: Design research in the technology of information systems: Truth or dare. Pennsylvania State University, Philadelphia (2002)
10. March, S., Storey, V.: Design Science in the IS Discipline – An Introduction to the Special Issue on Design Science Research. MIS Quarterly 32(4), 725–730 (2008)
11. Paradiso, J.: From Tangibles to Toolkits and Chaos toConvection–Management and Innovation at Leading Design Organizations and Idea Labs. In: Boland, R., Collopy, F. (eds.) Managing as Designing. Stanford Business Books, Stanford (2004)
12. Purao, S., Storey, V.: Evaluating the adoption potential of design science efforts: The case of APSARA. Decision Support Systems 44(2), 369–381 (2008)
13. Lindgren, R., Henfridsson, O., Schultze, U.: Design Principles for Competence Management Systems: A Synthesis of Action Research Study. MIS Quarterly 28(3), 435–472 (2004)
14. Carlsson, S.: Developing Knowledge through Is Design Science Research: For Whom, What Type of Knowledge, and How? Scandinavian Journal of Information Systems 19(2) (2007)
15. March, S., Smith, G.: Design and Natural Science Research on Information Technology. Decision Support Systems 15(4), 251–266 (1995)
16. van Aken, J.: Management research based on the paradigm of the design sciences: The quest for field-tested and grounded technological rules, pp. 219–246 (2004)
17. Markus, M., Majchrzak, A., Gasser, L.: A Design Theory for Systems That Support Emergent Knowledge Processes. MIS Quarterly 26(3), 179–212 (2002)
18. Henfridsson, O., Lindgren, R.: Multi-Contextuality in Ubiquitous Computing: Investigating the Car Case through Action Research. Information and Organization 15(2), 95–124 (2005)
19. Berndt, D., Hevner, A., Studnicki, J.: The CATCH Data Warehouse: Support for Community Health Care Decision Making. Decision Support Systems 35(3), 367-384 (2003)
20. Venable, J.: A Framework for Design Science Research Activities. In: Khosrow-Pour, M. (ed.) Emerging Trends And Challenges in Information Technology Management, Hershey, pp. 184–187 (2006)
21. McKay, J., Marshall, P.: The dual imperatives of action research. Information Technology and People 14(1), 46–59 (2001)
22. Davison, R., Martinsons, M., Kock, N.: Principles of canonical action research. Information Systems Journal 14(1), 65–86 (2004)
23. Lyytinen, K., Newman, M.: Explaining information systems change: a punctuated socio-technical change model. European Journal of Information Systems 17(6), 589–613 (2008)
24. Leavitt, H.: Applied organization change in industry: structural, technical, and human approaches. In: Cooper, S., Leavitt, H., Shelly, K. (eds.) New Perspectives in Organizational Research, pp. 55–71. Wiley, Chichester (1964)
25. Barros, A., Dumas, M.: The rise of Web service ecosystems. IT Professional 8(5), 31–37 (2006)
26. Strauss, A., Corbin, J.: Basics of qualitative research: Techniques and procedures for developing grounded theory, 2nd edn. Sage, Thousand Oaks (1998)

27. Baskerville, R., Pries-Heje, J.: Grounded action research: a method for understanding IT in practice. Accounting, Management and Information Technologies 9(1), 1–23 (1999)
28. Kirsch, L.: Portfolios of Control Modes and IS Project Management. Information Systems Research 8(3), 215–239 (1997)
29. Tiwana, A., Konsynski, B., Bush, A.: Platform Evolution: Coevolution of Platform Architecture, Governance, and Environmental Dynamics. In: Information Systems Research (20th Anniversary Special Issue) (2010)
30. Yoo, Y., Henfridsson, O., Lyytinen, K.: The New Organizing Logic of Digital Innovation: An Agenda for Information Systems Research. In: Information Systems Research (Information Systems Research 20th Anniversary Special Issue) (2010)

Extending Prior Research with Design Science Research: Two Patterns for DSRIS Project Generation

Bill Kuechler[1] and Vijay Vaishnavi[2]

[1] University of Nevada at Reno, Department of Information Systems, MS 026, 1664 N. Virginia Street, Reno, Nevada 89557, USA
kuechler@unr.edu
[2] Georgia State University, 35 Broad St., Room 902, Atlanta, GA 30303
vvaishna@gsu.edu

Abstract. Constructivist research – learning through building – is the core of a large stream of design science research in IS. Architecture has always explored through this paradigm; more recently, engineering-related disciplines, education and medicine have adopted it as well.

Constructivist methods are chosen in all cases because many systems problems are 'wicked': difficult, multi-faceted and frequently exhibiting aspects that emerge only during attempted solution of the problem. Constructivist methods excel at the investigation of incompletely understood problems where the variables of study are inextricably confounded or theory is sparse.

In this paper we present two patterns by which the power of constructivist methods can be directed at extending and generating practice-focused results from prior research for the benefit of the Information Systems discipline. The first pattern generates DSRIS projects based on theoretical findings; the second pattern generates DSRIS projects to clarify and extend poorly understood facets of large real-world artifacts/systems.

Keywords: information systems research methods, design science research.

1 Introduction

One view of design science research in IS (DSRIS) is as the application to information systems problems of the extremely powerful *constructivist research methodology*[1] – learning through building. The build-evaluate paradigm with the building sometimes guided only by intuition is the core of all constructivist methods. Constructivist research has been the primary method of exploration for the field of architecture for several thousand years; more recently, engineering-related disciplines, including computer science, and social systems designers in education and medicine have adopted constructivist research as well.

[1] In this paper the term *constructivist research* refers to *learning through building*. This is very different from its use in the social sciences where *constructivist research*, also known as *abstract constructivism,* refers to an approach to social analysis that deals with the role of human consciousness in social life.

H. Jain, A.P. Sinha, and P. Vitharana (Eds.): DESRIST 2011, LNCS 6629, pp. 166–175, 2011.

The inclination to constructivist methods is for similar reasons in all cases: many engineering, architectural and social systems problems are 'wicked' [1]: difficult, multi-faceted and frequently exhibiting emergent aspects that become visible only during attempted solution of the problem. Constructivist methods excel at the investigation of incompletely understood problems where the variables of study are inextricably confounded or have not yet been fully explicated by theoretical studies.

In this paper we present two patterns by which the power of constructivist methods can be directed at extending and generating practice-focused results from prior research in multiple fields for the benefit of the Information Systems discipline. The first pattern generates DSRIS projects based on as yet unapplied theoretical findings; the second pattern generates DSRIS projects to clarify and extend poorly understood facets of large real-world artifacts/systems or large academic information technology projects.

In the next section of the paper the strengths and benefits of constructivist methods are developed relative to the more statistically based methods found in positivist IS research. The primary emphasis is on the epistemology of IS constructivist research in order to emphasize the unique knowledge generation capabilities of the method. Next, the two DSRIS research project development patterns are presented. The first pattern generates DSRIS research projects from theoretical research, the second from aspects of large systems or constructivist projects. Extended examples of the use of each pattern are given in the appendices to this paper, online at: URL: http://www. weekspace.com/ anonymous/Generatingv2.0appendicies.pdf . Caveats for use of the patterns and opportunities for further development of a pattern approach to DSRIS research project generation are discussed in the concluding section.

2 Learning through Building: What Can Be Learned and How It Is Learned

A recent journal editorial described design science research as "the ancient but academically still emerging field. . . " [2]. While the exact nature and scope of design science research in information systems (DSRIS) is still a subject of discussion, we feel it is safe to state that: *many DSRIS projects are centered on the construction and evaluation of an IS artifact intended to improve a business process*. Unlike state-of-practice system DSRIS produces *new knowledge* by building artifacts with principles that are untried or seeks to exploit untested interactions between artifact components or both.

Figure 1 shows the steps in an idealized design science research project. One of the unique strengths of the constructivist method is illustrated by the knowledge flow arrows labeled *circumscription* [3]. Circumscription is the logical process by which constraints on the application of theoretical knowledge are recognized. In the constructivist method it is possible to explore theoretical possibilities in the evaluation context of the designed artifact; this is under the control of the design science researcher, to a degree not found in other methodologies.

Figure 2 is an abstraction of the knowledge generation process in the build-evaluate research cycle. We have modified the original diagram from Goldkuhl [4] with the addition of the dashed arrows and the elements in grey boxes so that the flow and derivation of knowledge matches the design research cycle of Figure 1. Another highly significant and unique capability of the constructivist method is apparent when sequentially tracing the arrows in Figure 2 labeled 1, 2, and 3.

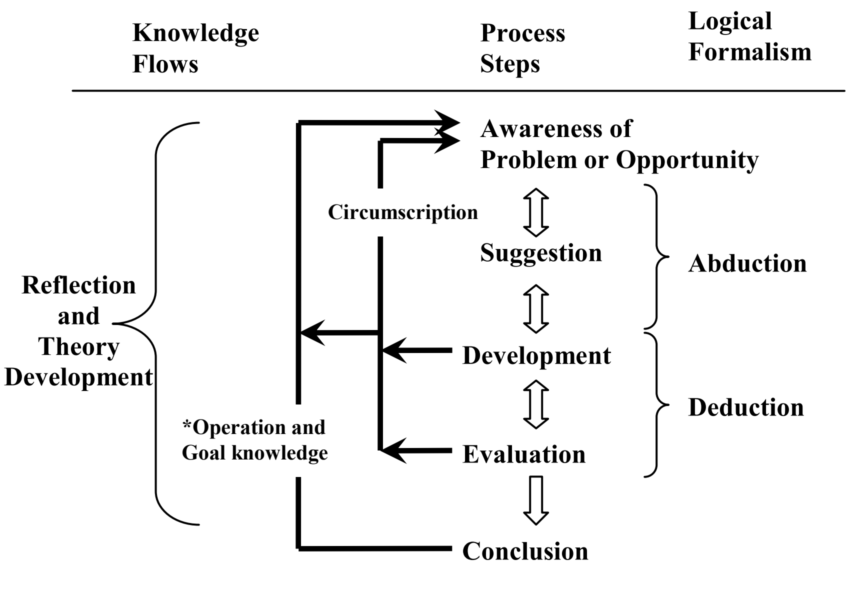


Fig. 1. Reasoning in the Design Research Cycle (From [5] as adapted from [6])

*An operational principle can be defined as "any technique or frame of reference about a class of artifacts or its characteristics that facilitates creation, manipulation and modification of artifactual forms." [7] [8]

This trace, when stated as a logical progression is as follows:

1. An artifact, constructed according to prescriptive statements – design rules, principles or paradigms – when evaluated, yields data ('Evidence' in Figure 2).
2. Consideration of the evidence leads to either:

   2.1 Confirmation of explanatory principles that may have motivated the design *or*

   2.2 Revision of the explanatory principles that may have motivated the design *or*

   2.3 Formation of a hypothesis of the mechanism of design functioning (explanatory principles) in the event that no explanatory principles had been formally derived prior to construction.

3. The new or revised explanatory statements lead to new or revised prescriptive statements – revised design principles, rules or paradigms.

Statement 2.3 above describes the ability of the constructivist method to literally bootstrap formal knowledge and/or design principles from intuition-guided problem solution construction. (This ability is one that we have in past writing termed 'atheoretic development', by which we meant that no formally stated principles informed the artifact construction).

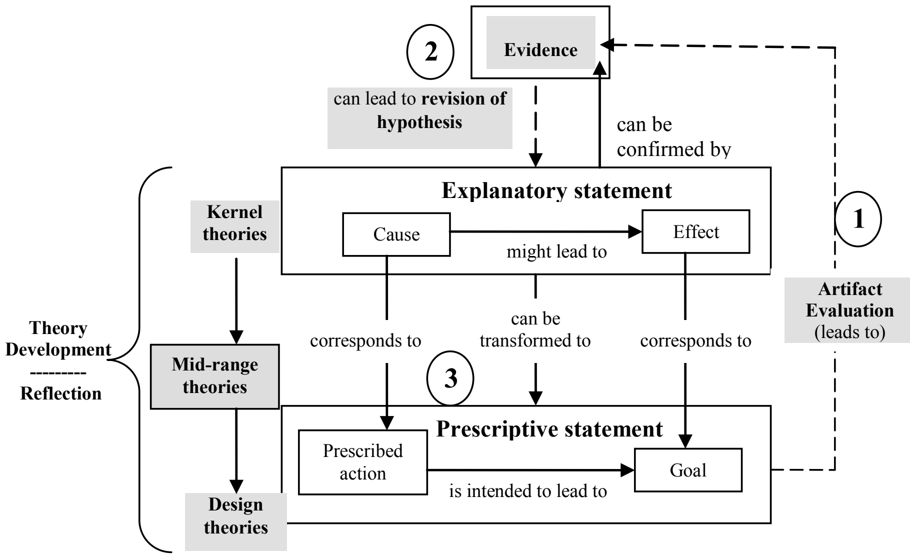


Fig. 2. Relationships between kernel theory, design theory, and the design process (modified from [4])

Patterns for generating design science research (DSRIS) projects from two different types of prior published research studies are given below. Pattern 1 generates projects to explore the potential benefits of translating as yet unapplied design-relevant theory to the design realm. This pattern corresponds to statements 2.1 and/or 2.2 above.

Pattern 2 generates narrowly scoped DSRIS projects to explore incompletely understood facets of much larger information technology development projects. The very scope of such projects frequently means that significant portions of them lie outside current theoretical understanding. Pattern 2 corresponds to statement 2.3 above. Initially there is no formally expressed understanding of the functioning of the artifact in its entirety.

3 The Patterns: Identifying DSRIS Opportunities

3.1 A Note on Pattern Description

Just as in software engineering (and earlier, in architecture [9]) we consider a **design pattern** to be a general reusable solution to a commonly occurring problem in design.

The DSRIS project generation methods we describe below certainly fit the definition of a design *pattern* for which the designed artifact is a research project. However, because these patterns for research project design are in a different realm and of a different order of complexity than software design patterns, we decided that the pattern description template familiar to readers of *Design Patterns* [10] was too confining. We have instead chosen to describe the patterns largely with narrative and with reference to Figures 1 and 2 above and Table 1 (below) to make the narrative more explicit. Examples of the exercise of each pattern can be found in on-line appendices to the paper, on line at: URL: http://www.weekspace.com/anonymous/Generatingv2.0appendicies.pdf

3.2 Pattern 1: 'Objectifying'/Translating Theoretical Studies

A precondition for application of this pattern is that during the *Problem Definition* or *Suggestion* phases of the design research cycle (Figure 1) a published theoretical research study has been discovered that is highly congruent with an interesting aspect of the DSRIS business problem. By that we mean the theoretical research (uncovered during the literature search portion of one of the early phases of the design research cycle) immediately (or with a modest amount of reflection) suggests a design approach to the problem. In our experience this type of serendipity is not uncommon when exploring systems interface problems; in that area congruent theoretical research can be found in journals of human-computer interface studies and applied and educational psychology journals among others.

The seminal papers on DSRIS design theory (DT) by Walls, et al. [11] [12] and the many papers that use and reference DT refer to *kernel theory* as a component of the DT. The kernel theory, frequently from a natural sciences field [13] [12], describes the principles that suggest a novel design approach to the DSRIS project problem. Yet none of the published work describing, using or generating design theory explicitly sets forth the means by which the kernel theory concepts are mapped to the DSRIS project facets. The pattern described next details the translation of insights, dependent variables (DVs) and independent variables (IVs) from kernel theory to more concrete environments related to the IS problem being solved or technology being improved as the focus of the DSRIS project. Once translated, DV(s) and IV(s) can be explored in a near-to-use context (DSRIS evaluation) yielding design knowledge and exposing limits on the kernel theory through circumscription (see Figure 1, *Reasoning in the Design Research Cycle*).

To introduce the pattern at a conceptual level, consider the IS problem: *search engines are imprecise*. Whether the issue is approached as a business problem – lost time and money in information searches – or as a technology issue – how can web search technology be improved – the DSRIS solution always eventually points to a technology, which is the IV to be manipulated in some way. The DV is the evaluation measure; to continue with the search engine example, the DV could be money or time or precision as a surrogate for efficiency (money, time). The manipulation of the problematic technology is suggested by the kernel theory.

For example suppose a theoretical HCI research project determines that there is an optimum number of on-screen display elements humans can attend to. The number depends on size, complexity, etc. The theoretical research stops at this very general conclusion which provides no specific guidance to designers of browser-based web-search interfaces. The kernel theory suggests that its IV influences its DV. The kernel theory IV must map by analogy or some chain of reasoning to a manipulable element in the DSRIS project technology. The kernel theory DV must map to the DSRIS project artifact DV, the evaluation measure(s).

Translation from kernel theory DV/IV to DSRIS project variables requires two translations, first from theoretical abstraction to a specific aspect of the technology and second from technology aspect to specific design for that aspect. As we describe the steps in the pattern we will relate them to phases in the general design research cycle (Figure 1). With respect to Figure 1, once a theoretical study of interest has been identified, it becomes necessary to identify a specific technology and a specific problem or shortcoming with that technology (a 'facet') for which the theory suggests a solution. At that point both the Awareness of Problem phase and the Suggestion phase of the design cycle (Figure 1) have been preliminarily completed. Translation 1 (below) takes the theoretical implication from the study and specializes it for the identified technology facet.

Translation 1, a conceptual translation - the mapping of kernel theory DVs and IVs to a specific technology facet which we desire to improve. As mentioned above, this mapping takes place during the suggestion phase of the design research cycle (Figure 1). The technology used in the DSRIS project may be suggested or even completely determined by the apparatus or artifact used to develop the kernel theory. Continuing with our web-search example, assume the theory was developed through observation of users of web-browser-based search engines. Given this and the nature of our research question (how can a better interface improve query performance), the DSRIS technology base artifact may well also be a browser. The DSRIS research project IV will be a specific approach to the "problem" or deficiency targeted for the technology. Let us assume the DSRIS researcher proposes that the difficulty users encounter in interpreting the mass of search results returned contributes to the inefficiency of the composite system – the search engine, its interface and the user. The kernel theory suggests that there are significant differences in the ease of interpreting screen displays, and so the specific, technological IV for the project is the *results display for a search*.

Table 1. Logical form and semantics for mapping theoretical concepts into a design domain

Kernel theory construct / proposition	Mapping	Technology construct / proposition	Semantics
X (construct; dependent variable)	➔ (is equivalent or analogous to)	Y (technology constrained construct; dependent variable)	Kernel theory construct X (or concept) maps (is analogous) to technology (artifact) construct Y
B (treatment or independent variable)	➔	D (artifact performance variable)	Kernel theory treatment variation B is analogous to artifact-in-use action D
B acting on X causes C	➔	Do D to Y to get result E (Design Theory as Design Rule)	Thus, since performing B on X causes C then artifact activity D performed on Y will yield effect E

Translation 2, a design translation - in the generation of a DSRIS project from a kernel theory is the *concrete* translation of DV and IV into the design domain, that is, into a specific design rule set for aspects or features of the designed artifact. The logic underlying the translation process is shown in Table 1. This translation takes place during the Development[2] and Evaluation phase of the design research cycle (Figure 1) and is frequently iteratively refined in cycles through these phases or even branching back to Step 1 of this method.

The question that drives the mapping/translation from theory IV to design is: "In what physically executable form can the IV be expressed in the technology implementation?"

Continuing with our example, the DSRIS researcher may propose a design that (1) formats returned search results in a specific way and that (2) allows users to specify the exact number (within theoretically suggested guidelines) of results displayed. This design can be expressed in general rules.

To review Pattern 1, our supposed theoretical HCI research project determined that there is an optimum number of on-screen display elements humans can attend to. The number depends on size, complexity, etc. The theoretical research stops at this very general conclusion which provides no specific guidance to designers of browser-based web-search interfaces. To construct a DSRIS project(s) with Pattern 1:

1. Map the theory to a technology – we assumed a browser.
2. Map the theory DV and IV to a technology facet (Translation 1). Impose context on the display: in our example, let us assume a browser used for search when results returned consist of some narrative + URL (similar to Google's behavior). This is the IV. The DV is improved search performance.
3. Design the novel or improved facet (Translation 2, above). For the example, determine the optimum display characteristics (the independent variable, IV) as measured by the DSRIS dependent variable (DV), user behavior, and express them as a design rule or pattern (design theory).

Note that even with the guidance of the pattern some of the activities of some of the pattern steps require the exercise of creativity. This is most notable in step 3 (immediately above) during which design rules emerge. However, the guidance provided by the pattern both circumscribes and focuses the creative effort. An example of the exercise of this pattern using a published theoretical paper from HCI is given in Appendix 3 to this paper, on-line at: URL: http://www.weekspace.com/anonymous/Generatingv2.0appendicies.pdf

3.3 Pattern 2: Deconstructing Large Information Technology Projects

Many information technology projects are huge by DSRIS standards. Some examples are MOOsburg [14], a precursor of contemporary on-line, interactive virtual environments, the 'smart home' project [15], or even full-featured computer language tutorial systems [16]. The method suggested by Carroll and Rosson [14] for

[2] The *Development* phase of the design cycle of Figure 1 includes both *design* and *implementation* of the artifact.

explicating explicit design knowledge from HCI projects is an excellent method to begin to decompose a large information technology project into smaller DSRIS projects. Carroll and Rosson decompose a large, complex artifact into features; underlying each feature is a *claim* for the effective functioning of the feature. Claims are generated through *claims analysis*: a causal analysis of the users' experience with the artifact.

Each claim is typically based on a grounding 'natural science' theory from psychology or other behavioral science or possibly from a design theory from IS or computer science. The claim/theory constitutes a strand in a web of claims for the artifact that Carroll and Rosson [14] term the *design rationale* for the larger artifact[3].

The theory underlying each claim is equivalent to a DSRIS kernel theory. The claim is **for** a feature of a specific artifact, which is to say both the kernel theory and the implementation have *already been concretely scoped in the course of the design and construction of the artifact*. The DVs and IVs of the feature-underlying theory have necessarily been mapped to the design domain. The behavior of any feature of the artifact exercises the IV(s) and the user reaction (behavior) or artifact (or aspect thereof) performance is the DV(s). A DSRIS project can be generated from each claim to investigate the feature in a more controlled manner, creating a design theory and potentially extending the understanding of that feature's kernel theory.

The DSRIS project can be thought of as a piece of the larger project that has been "broken off" to isolate and investigate more deeply the DVs and IVs of the feature's theory *and* the effectiveness of the translation of these to design. The DSRIS project might explore broader variations of the IV than was done in the larger artifact, different constraints on the user experience than in the larger artifact and/or alternate design translations of the variables – different ways of varying the IV in an artifact of similar intent (functionality) to the original.

The results of each DSRIS exploration are an exemplar artifact, and a DT expressed either in formal terms or as design patterns and potentially *increased understanding through circumscription* of the original theory, which can be captured in the form of mid-range theory (see Figure 2 and the discussion of DSRIS project outputs in the prior section).

We assume the motivation is that there are, on reflection, meaningful unanswered questions about the use of the artifact feature or its capabilities or scope. Treat the claim as the kernel theory, conceptualize an artifact that probes the unanswered question and proceed as per Pattern 1. To the degree the new technological artifact is related to the larger artifact from which the kernel theory was taken, the design information for that larger artifact may be useful.

Summarizing Pattern 2 steps overall:

- Deconstruct a large IT artifact/system into features
- For each feature of interest, identify the behavioral or performance claims for that feature
- Mark the claims as beneficial (+) to the feature, neutral or detrimental (-)
- The (-) claims specify a likely area for improvement.

[3] Design rationale may operate at different levels. "Design rationale for families of artifacts also overlaps, leading to more abstract design rationale that create a design space." [14], p. 434.

- The (+) claims may be amenable to enhancement
- Next identify the theory that generated the claim for the design feature.
- Identify the DV and IV for the theory as shown in Table 1, above.
- Next conceptualize a DSRIS artifact construction project that could explore the theory by varying the artifact implementation of the feature
- Note that the interplay of (-) and (+) claims necessitated a design compromise. Mitigating (-) claims or enhancing (+) claims will probably change the design rules for the feature and for the larger artifact/system

For a specific example of artifact deconstruction into features, claims and underlying theories in an HCI context see [14], figures 15.3 and 15.4. An example of the exercise of this pattern to deconstruct a large IT project into a feature-focused DSRIS project is given in the appendices to this paper, on line at URL: http://www.weekspace.com/anonymous/Generatingv2.0appendicies.pdf

4 Concluding Remarks, Limitations of the Research and Caveats

In this brief paper we have presented two patterns for generating DSRIS projects. The first pattern translates design-relevant theory from any domain into the IS design domain. The second pattern describes how large artifacts or systems generated in IT practice or by large academic IT projects can be deconstructed into facets. Poorly understood facets of interest to real-world IT problems can be explored through the DSRIS projects generated by this pattern. Another contribution of the paper is the DSRIS project description template of Appendix 1 (on line at URL: http://www.weekspace.com/anonymous/Generatingv2.0appendicies.pdf which may find broader use.

The primary caveat we offer on the research is that the patterns developed here focus on 'traditional', 'hard' DSRIS artifacts: software or systems. The patterns may be amenable to the development of IS process artifacts also but this possibility has not been treated in the paper.

Many more patterns for constructing IS design science research projects await development and this is one of our future research goals. Indeed, an entire book has already been devoted to DSRIS patterns [17]. However, we feel the patterns presented in this paper are more detailed and sophisticated than any previously published.

Finally, we have been able, within the confines of a conference paper and the online appendices, to present only a single example of the use of each pattern, and the projects suggested by the examples have not been enacted. However, we believe the patterns have high face validity as do the projects suggested in the examples of use. In fact, the patterns we have described emerged largely through our reflection on the process and ratiocination of a number of past DSRIS projects in which we have been involved. One such previously published project has been retroactively explicated using Pattern 1 and is presented as an additional example of the use of that pattern in Appendix 2 (on line at URL: http://www.weekspace.com/anonymous/Generatingv2.0appendicies.pdf

References

1. Rittel, H., Webber, M.: Dilemmas in a General Theory of Planning. Policy Sciences 4, 155–169 (1973)
2. Koskinen, I.: The Editor's Introduction: Design Research from Northern Europe, Part II. Knowledge, Technology & Policy 20(3), 129–130 (2007)
3. McCarthy, J.: Circumscription - A Form of Non-Monotonic Reasoning. Artificial Intelligence 13, 27–39 (1980)
4. Goldkuhl, G.: Design Theories in Information Systems - A Need for Multi-Grounding. Journal of Information Technology Theory and Application 6(2), 59–72 (2004)
5. Vaishnavi, V., Kuechler, W.: Design Research in Information Systems (2004), `http://desrist.org/design-research-in-information-systems/` (accessed January 20, 2004, last updated August 16, 2009)
6. Takeda, H., Veerkamp, P., Tomiyama, T., Yoshikawam, H.: Modeling Design Processes. AI Magazine 11(4), 37–48 (1990)
7. Purao, S.: Design Research in the Technology of Information Systems: Truth or Dare, GSU Department of CIS Working Paper. Atlanta, GA (2002)
8. Dasgupta, S.: Technology and Creativity. Oxford University Press, New York (1996)
9. Alexander, C., Ishikawa, S., et al.: A Pattern Language: Towns, Buildings, Construction. Oxford University Press, Berkley (1977)
10. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns. Addison-Wesley, Upper Saddle River (1994)
11. Walls, J., Widmeyer, G., et al.: Building an Information System Design Theory for Vigilant EIS. Information Systems Research 3(1), 36–59 (1992)
12. Walls, J., Widmeyer, G., et al.: Assessing Information System Design Theory in Perspective: How Useful was our 1992 Initial Rendition. Journal of Information Technology Theory and Application 6(2), 43–58 (2004)
13. March, S., Smith, G.: Design and Natural Science Research on Information Technology. Decision Support Systems 15(4), 251–266 (1995)
14. Carroll, J., Rosson, M.: Design Rationale as Theory. In: Carroll, J. (ed.) HCI Models, Theories and Frameworks: Toward a Multidisciplinary Science, pp. 431–461. Morgan Kaufmann, San Francisco (2003)
15. Kidd, C., Orr, R., Abowd, G., Atkeson, C., Essa, I., MacIntyre, B., Mynatt, E., Starner, T., Newstetter, W.: The Aware Home: A Living Laboratory for Ubiquitous Computing Research. In: Second International Workshop on Cooperative Buildings - CoBuild 1999 (1999)
16. Carroll, J., Rosson, M.: Getting Around the Task-Artifact Cycle: How to Make Claims and Design by Scenario. ACM Transactions on Information Systems 10(2), 181–212 (1992)
17. Vaishnavi, V., Kuechler, W.: Design Science Research Methods and Patterns: Innovating Information and Communication Technology, New York, Auerbach (2008)

Inductive Design of Maturity Models: Applying the Rasch Algorithm for Design Science Research

Gerrit Lahrmann[1], Frederik Marx[1], Tobias Mettler[2], Robert Winter[1], and Felix Wortmann[1]

[1] Institute of Information Management, University of St. Gallen, Müller-Friedberg-Str. 8, 9000 St. Gallen, Switzerland
{gerrit.lahrmann,frederik.marx,robert.winter, felix.wortmann}@unisg.ch
[2] SAP (Switzerland) AG, SAP Research CEC St. Gallen, Blumenbergplatz 9, 9000 St. Gallen, Switzerland
tobias.mettler@sap.com

Abstract. Maturity models are an established means to systematically document and guide the development of organizations using archetypal capability levels. Often, these models lack a sound foundation and/or are derived on the basis of an arbitrary design method. In order to foster the design of relevant and rigorous artifacts, this paper presents a method for maturity model construction that applies the Rasch algorithm and cluster analysis as a sound methodical foundation. The Rasch algorithm is widely used to improve scholarly intelligence and attainment tests. In order to demonstrate the application of the proposed method and to evaluate its usability and applicability, we present a design exemplar in the business intelligence domain.

Keywords: design science, maturity models, Rasch algorithm, business intelligence.

1 Introduction

In the field of Information Systems (IS) and Management Science, maturity models (MM) are steadily being applied both as an informed approach for continuous improvement [1, 2] and as a means for self or third party assessment [3, 4]. Since the introduction of the concept in the 1970s [e.g. 5, 6], a multiplicity of different MMs has been developed in academia and practice, making it an excellent example for design research [7]. Becker et al. [8] report that more than 1,000 articles refer to MMs and Mettler et al. [9] found more than 100 MMs.

Nonetheless, the concept has not been undisputed by criticism. For instance, Pfeffer & Sutton [10] argue that the purpose of MMs is to identify a gap, which can then be closed by subsequent improvement actions. However, lots of these models do not describe how decision makers effectively have to perform these actions. Hence, often a 'falsified certainty of success' is purported. Mettler and Rohner (2009) excoriated that yet today situational factors (e.g. corporate culture, organization structure, company size) are often left behind in favor of reducing complexity (thus assuming a static and somewhat simplistic world view). In the business intelligence

H. Jain, A.P. Sinha, and P. Vitharana (Eds.): DESRIST 2011, LNCS 6629, pp. 176–191, 2011.

(BI) domain, a recent methodical state-of-the-art analysis reveals that MMs often lack a theoretical foundation, are not well documented, and especially dismiss methodical requirements [11]. This is in line with Biberoglou & Haddad [12], who state that there exists no common and widely accepted design methodology and formal theoretical basis for MM construction so far.

The goal of this paper is to address the core issue outlined by Lahrmann et al. [11] and Biberoglou & Haddad [12]. Hence, the paper proposes a psychometric-driven approach for designing relevant *and* rigorous MMs based on the Rasch algorithm and cluster analysis. Advancing the knowledge base of artifact construction, this paper aims at contributing to the meta research of design science [13].

Following a basic design science process of "build" and "evaluate" [14, 15], the paper is structured as follows. In section 2, we outline the methodical foundations as well as theoretical considerations for the construction of MMs. In section 3, the underlying assumptions and our algorithm-based MM construction methodology are discussed in detail. In order to clarify and evaluate the suggested construction method, we develop an exemplary BI MM in section 4.[1] Finally, we summarize our findings and offer suggestions for future work.

2 Foundations

2.1 Definition of Maturity and Demand for a Maturity Model

In general, 'maturity' can be defined as "the state of being complete, perfect or ready" [16]. Maturity thus implies an evolutionary progress in the demonstration of a specific ability or in the accomplishment of a target from an initial to a desired or normally occurring end stage. In the IS discipline, 'maturity' is rather regarded as "a measure to evaluate the capabilities of an organization" [17]. Following Becker et al. [18], MMs endorse this evaluation by outlining anticipated, typical, logical, and desired *evolution paths*. Furthermore, Mettler & Rohner [19] argue that, as formality is incorporated into the organizational development activities, decision makers are given a pragmatic instrument to determine whether potential benefits have been realized or not.

To delineate this evolution path, either a top-down or a bottom-up approach is conceivable [20]. With a top-down approach, a fixed number of maturity stages or levels is specified first and further corroborated with characteristics (typically in form of specific assessment items) that support the initial assumptions about the maturity distribution. When using a bottom-up approach, distinct characteristics or assessment items are determined first and clustered in a second step into maturity levels to induce a more general view of the different steps of the evolution path. In this sense, it is possible to define a different scale, number and type of maturity level for each of these clusters or *focus areas* [21].

Independent of whether a top-down or bottom-up approach is applied for describing the evolution path, reflections with respect to uncertainty of this evolution path have to be made [22]. In this regard, theories of emergence and diffusion of innovations reveal interesting foundations. According to Utterback & Abernathy [23],

[1] It is not the motivation of this paper to present an in-depth discourse on future BI-trends, but rather to show how the Rasch algorithm can be applied for maturity model construction.

the progress of a particular innovation follows an S-curve (left side of Fig. 1). Innovation (mainly originating from many minor product or process improvements) passes through different stages of maturity. Of particular interest is thereby the disruptive phase where a dominant design of a solution (i.e. a general agreed standard or best practice) becomes evident. However, dominant designs may not be better than other designs, but acceptance of an innovation will be on peak (right side of Fig. 1).

As regards the development of MMs, the cognition of the state in which an innovation is situated is thus extremely important, especially when the model is prescriptive. For instance, when focusing on the development of a MM for an emerging phenomenon, the levels of maturity may be extremely uncertain given that no dominant design is found already. Furthermore, only limited 'test cases' are available for effectively observing the accuracy of the recommended improvement actions, thus reducing the possibilities for attaining a reliable sample. Accordingly, this may have an effect on the appreciation of the MM as such, since the described evolution path may be perceived as speculation. On the other hand, when concentrating on a mature domain with lots of 'test cases' to rely upon, the levels of maturity may be clearer. However, the utility of the model may be reduced since the demand for guidance is lower. In such a case the results from an appraisal may be understood as 'bureaucracy' or 'platitude' as no substantial benefits may be gained.

A similar train of thoughts can be made when considering the diffusion of innovations. When using too fundamental or forward-looking criteria for the maturity assessment of an organization, the application of the model will show an accumulation of the results on a predefined sophistication level. For instance, Hayes & Zubrow [24] discovered that 73 percent of the assessed organizations between 1987 and 1994 were stuck on the first level of the Capability Maturity Model (CMM) because the requirements of the process area 'project management' were far too hard to meet. Therefore when defining the levels of maturity for a particular domain, a trade-off between the state of an innovation's uncertainty and its actual diffusion (which assists in predicting whether and how an innovation will be successful) has to be considered in order to guarantee 'useful insights' (i.e., trustworthy but not too obvious improvement activities) from the application of the model.

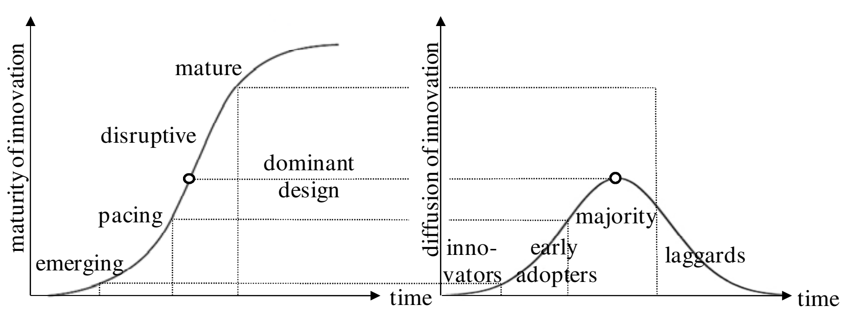


Fig. 1. Relationship between Maturity and Diffusion. Adapted from [22].

2.2 Maturity Model Development Methods

In contrast to other artifacts of the type "model", e.g. reference process models [25], only limited knowledge is available on how to systematically delineate the evolution path and how to instantiate the corresponding MM. As for now, five distinct development processes have been extensively discussed in current literature [18, 20-22, 26][2]. All of these processes share five generic design steps: (1) *Identify need or new opportunity*: Developing MMs by conducting design-oriented research means finding solution patterns for important unsolved problems or giving advice in solving problems in more effective or efficient ways [15]. According to Järvinen [27], a business need is not necessarily required but a new opportunity as "opportunity-based innovation can have a great economic value". As discussed in the prior section, the demand for a new MM strongly depends on the diffusion and maturity of the subject itself, given that mature themes typically require less explanation than emerging ones. (2) *Define scope*: In order to develop a useful model, the domain must be scoped properly [21]. This means that it has to be decided whether certain assumptions and characteristics are included or not. According to de Bruin et al. [20], this will "set the outer boundaries for model application and use". (3) *Design model*: The artifact as such is constructed in a third step. Becker et al. [18] and de Bruin et al. [20] both suggest a top-down approach by primarily defining a kind of grid or architecture of the relevant domain dimensions and sub-dimensions and 'filling' these dimensions with typical characteristics using focus groups/delphi method, creativity techniques, case studies, or literature reviews. Quantitative methods are less frequently used for constructing MMs. An example is the Rasch algorithm-based approach [22, 28], which is discussed and enhanced in this paper. (4) *Evaluate design*: Due to the fact that the acceptance of a MM critically depends on its utility, validity, reliability, and generalizability, evaluation is a crucial step in every design science research project [15]. (5) *Reflect evolution*: Finally, the MM has to be maintained and further development will be needed given that some model elements will get obsolete, new constructs will emerge, and assumptions on the different levels of maturity will be affirmed or refuted [18]. Therefore, even in an early stage it is important to also reflect on how to handle alterations in model design and deployment. However, this design step has attained little attention yet. Table 1 provides an overview of the applicable development methods and their adequacy for MM construction.

3 Using the Rasch Algorithm for MM Construction

3.1 Background and General Assumptions

Dekleva & Drehmer [29] favor the usage of the Item Response Theory (IRT) in the context of MMs. IRT has been developed to overcome deficiencies with classical test theory. Both theories aim to improve the reliability and validity of standardized tests or questionnaires by deriving conclusions about difficulty of items or the ability of participants [30]. IRT assumes that responses to questionnaire items are dependent on non-measureable respondent characteristics (a single latent variable or trait θ) and on item characteristics. Respondent characteristics could be psychological traits, but also

[2] For a more detailed analysis of the distinct development methods, please refer to [21, 27].

Table 1. Maturity Model Development Methods

Phase	Design method	Exemplary Source
1. Identify need or new opportunity	• Creativity techniques • Focus groups • Case studies • Literature review • Survey	• (Becker, et al. 2009) • (Mettler 2010a) • (de Bruin, et al. 2005) • (Van Steenbergen, et al. 2010)
2. Define scope	• Informed arguments • Scenarios	• (Van Steenbergen, et al. 2010) • (Mettler 2010a)
3. Design model	• Top-down: ▪ Delphi method ▪ Case studies ▪ Literature review • Bottom-up: ▪ *Algorithmic analysis* ▪ Informed arguments ▪ Ontologies	 • (Becker, et al. 2009) • (De Bruin et al. 2005) • (Van Steenbergen, et al. 2010) • (Mettler 2010a) • (Mettler 2010a)
4. Evaluate design	• Functional testing • Structural testing • Survey • Focus groups • Interviews	• (Mettler 2010a) • (Mettler 2010a)
5. Reflect evolution	• Field study • Interviews	

knowledge and specific capabilities. Determining a latent trait, IRT is a probabilistic alternative to factor analysis.

By basically counting properly answered questions, IRT calculates a score for the difficulty of items β_i and the ability of participants θ_v. Both scores are measured on the same interval scale. In the context of MM construction, the measurement of item difficulty supports the inductive allocation of items onto maturity levels and the capability of participants supports the assessment of organizations. The basic assumption and principle is schematically outlined in Fig. 2, where the responses of a dichotomous questionnaire are ordered in a response matrix. The sum scores are used to order the capabilities and difficulties.

Treating raw scores directly as measurements for the difficulty of items and capabilities of organizations, as in Fig. 2, does not yield in adequate results. Such a procedure tends to concentrate around middle scores and does not contrast the results of the more capable and less capable organizations, e.g. a very difficult item has the same contribution to the score as an easy item [30]. Therefore, the raw scores are first converted in their success-to-failure ratio, which allows displaying items and abilities ordered on one scale. Secondarily, the values are converted to their natural logarithm which calibrates the difficulty and ability around zero and therefore the score from a merely ordinal scale to interval scale, a log odds unit scale (see Table 2). A logit of zero corresponds to average difficulty (or ability). An additional logit doubles, a subtraction bisects difficulty or ability [31].

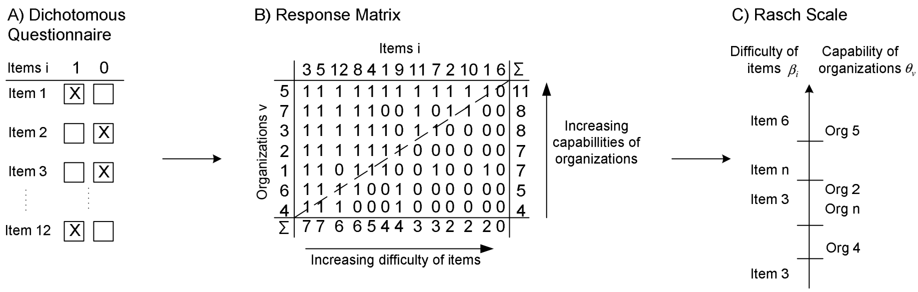


Fig. 2. Schematic Principle of the Rasch Algorithm

Furthermore, raw scores do not recognize patterns of responses by organizations. For example in Fig. 2, the raw score of organization 2 is orderly, as all easy items are properly answered and all difficult items are not. But it would be unfair to treat it similarly to organization 1 with the same raw score, but a pretty mixed response pattern; the same is true for the difficulty of item 10 compared to item 1. Therefore, using the Rasch algorithm, the capabilities and difficulties are tested against estimates based on a item response function (IRF) [32]. The IRF is based on the nonlinear monotone homogeneity model (MHM) which fits to the basic consideration of MMs in section 2.1. It states that highly skilled organizations have a high probability of having successfully implemented easy characteristics. Similar, low skilled organizations have a low probability of using advanced practices. The IRF is defined as the conditional probability of success $f_i(\theta_v)$ for an organization v to solve item i. Consequently, the calculation of estimates of difficulty and capabilities is based on the inverse IRF using Maximum Likelihood Estimates (MLE) [32].

The degree of difficulty of an item, the ability of organizations θ_v, the IRF, and estimates can be calculated by the following steps presented in Table 2 [33].

Table 2. Calculation of the Measures, IRF and Estimates

Step	Formula	Description
0	-	Eliminate all items, which have been solved by all or none of the organizations, as no additional information is provided.
1	$\beta_i = \ln\frac{1-p}{p}$;	Calculate degree of difficulty of an item *i*: share of not or incorrect answered items (1-*p*) divided by share of correct answered items (*p*).
2	$\theta_v = \ln\frac{p}{1-p}$;	Calculate ability of an organization v: share of correct answered items (*p*) divided by share of not or incorrect answered items (1-*p*).
3	$f_i(\theta_v) = \mathrm{P}(\mathrm{X_{vi}} = 1 \mid \theta_v, \beta_i) = \frac{\exp(\theta_v - \beta_i)}{1+\exp(\theta_v - \beta_i)}$	Calculate the conditional probability of success $f_i(\theta_v)$. Using one scale, it is based on the difference between the ability of organizations θ_v and the difficulty of the items β_i, so that for a given value θ_v the probability decreases with increasing difficulty β_i.
4	$\hat{\theta}_v, \hat{\pi}_{r_v i}$	The estimates of difficulties and items are derived by using MLE based on the inverse IRF.

The Rasch algorithm is able to handle binary scales and rating scales [30]. For simplicity reasons, the paper provides a short introduction using binary scales first.

Key indices for testing are OUTFIT and INFIT [32]. Both are based on chi-square fit statistics using the residuals between the observed and the expected value $\hat{\pi}_{r_v i}$ under the IRF (see Table 3). They indicate how accurately or predictably data fits the MHM.

Table 3. Calculation of OUTFIT and INFIT indices

Formulary	Description	Expectation
$OUTFIT_i = \frac{1}{V}\sum_{v=1}^{V}\frac{(x_{vi}-\hat{\pi}_{r_v i})^2}{\hat{\pi}_{vi}(1-\hat{\pi}_{vi})}$	OUTFIT is based on the conventional sum of squared standardized residuals. It is more sensitive to extreme scores.	Expectation of 1, values substantially below 1 indicate dependency in data; values substantially above 1 indicate noise.
$INFIT_i = \frac{\sum_{v=1}^{V}(x_{vi}-\hat{\pi}_{r_v i})^2}{\sum_{v=1}^{V}\hat{\pi}_{vi}(1-\hat{\pi}_{vi})}$	INFIT is an information-weighted sum. The statistical information is its variance. This is larger for targeted observations, and smaller for extreme observations. So it is inlier-sensitive.	Expectation of 1, values substantially below 1 indicate dependency in data; values substantially above 1 indicate noise.

For details and mathematical specifications of the underlying assumptions, the specialized Rasch literature is recommended [30-32, 34].

3.2 Rasch Algorithm-Based Maturity Model Construction

The MHM and the positioning of difficulty of items and ability of organizations fit to the basis considerations of MMs, which has already led to applications of Rasch for MM construction by selected researchers [22, 28]. To make it applicable for MM construction in the field of IS research, the basic model needs three modifications, without challenging the basic assumption of the Rasch algorithm:

First, in the context of complex socio technical systems the expressive power of rating scales, e.g. a Likert scale from one to five, is superior to dichotomous scales. A Likert scale clearly recognizes that the questions are requiring merely expressed opinions than just simple right or wrong answers [35]. Therefore, rating scales instead of dichotomous scales should be deployed. The Rasch algorithm is already prepared for the handling of rating scales. But the determination of the characteristics is slightly more complex, for details refer to the specialized Rasch literature [30, 34].

Second, in order to develop MMs in the context of IS, researchers should not only ask for the actual situation of an item i at organization v A_{vi}, but also for the desired situation of item i at organization v D_{vi}. Following the principle of economic efficiency [36], the overall utility function of an item i at organization v is not necessary monotonically increasing, but could be limited by D_{vi} providing an upper bound for the item. The potential and desired improvement is then provided by the delta value between desired and actual values $X_{vi} = D_{vi} - A_{vi}$. A negative delta expresses overcompliance or undesired developments. The delta value X_{vi} can now serve as the input variable for the Rasch algorithm. Considering that MMs should provide an individual as-is assessment, but a common and consistent development perspective, this paper proposes the usage of a modified delta between the individual actual values A_{vi} (organization specific) and common desired values $\widetilde{D}_i$ (median across all organizations) $X_{vi} = \widetilde{D}_i - A_{vi}$. The median $\widetilde{D}_i$ is able to express the importance of one item for all organizations. Compared to the arithmetic mean, the median is much more stable for extreme values [30]. Thus, the median is preferred over the mean. The difficulty of an item per organization is then given by the delta value: A high positive gap expresses a difficult and desired item. Negative gaps and also values on the threshold itself express easy items. Following common practice of Rasch model application, categories are collapsed with respect to the situation at hand [30]. Therefore, all negative gaps are collapsed into one category, as they all express overcompliance. Also very large gaps are collapse into one category as there is hardly any distinction between "very, very" difficult items and "very, very, very" difficult items. Therefore, a re-coding of data is applied resulting in five values. The result is presented in Table 4. Following Rasch model application for Likert scales, more easy items are expressed with a higher value.

Table 4. Coding of Actual and Desired Values

Delta Δ	<0	0	1	2	>2
X_{vi}	5	4	3	2	1
Interpretation	Easy items			Difficult items	

Third, the Rasch algorithm only yields a single ordinal scale that represents the logit measure of each item and organization, but not distinct maturity levels. In order to overcome subjectivity in defining maturity levels (e.g. by defining level one items to have logits smaller than -1), we propose the usage of cluster analysis on the basis of the item logits. The purpose of clustering is to investigate "a set of objects in order to establish whether or not they fall […] into groups […] of objects with the property that objects in the same group are similar to one another and different from objects in other groups." [37] Within various clustering methods, agglomerative algorithms have the largest significance in practice and are selected here for maturity level distinction [38]. Based on the measure of the items, they start with n clusters, each containing a single item. Using a standardized distance measure, step by step the number of clusters is reduced. As most MMs use five maturity levels [8, 11], the anticipated number of clusters is set to five.

4 Evaluation of the Method

In order to show the applicability of the method proposed, we outline a design exemplar for a BI MM. Section 4.1 provides an overview of the survey design and sample. Section 4.2 presents the results of the Rasch algorithm and the cluster analysis. Based on these results, section 4.3 reflects on the method presented at hand.

4.1 Survey Design and Data Collection

For the design exemplar, a questionnaire is used that is structured into four dimensions. Following well accepted management and IS frameworks, three functional BI dimensions of “strategy” (“what”), the “organization and processes” (“how”) and the “IT support” (“with what”) are differentiated [39]. Additionally, the fourth dimension “quality of service” focuses on how well the BI organization performs in its environments. Therefore, it contains questions related to non-functional aspects such as user satisfaction, performance and scalability [40]. For each dimension, two up to six questions are defined, resulting in a total of 14 questions. Each question contains two up to six characteristics, which are assumed to have different levels of difficulty, respectively maturity. This results in a sum of 58 items. A paper-based questionnaire is used to collect data. The responses are scored on a five-point Likert scale. The participants of an international conference for practitioners in the field of BI held in 2010 in Switzerland served as sample. From the 144 participants of the conference, 51 returned the questionnaire yielding in a response rate of 35.4%. In Table 5, the characteristics of the sample are outlined.

Table 5. Sample Characteristics

Industry sector	No	%	Employees	No	%
Automotive	2	4	0-100	9	18
Chemical & pharmacy	3	6	100-1000	11	22
Services	13	25	1000-10000	13	25
Utilities	3	6	> 10000	13	25
Finance & banking	8	16	Not available	5	10
Healthcare	1	2	Sum	51	100
Wholesale & retailing	4	8	**Position of respondent**	**No**	**%**
Techn., new media & telecom.	5	10	Business organization	5	10
Transportation & logistics	2	4	IT organization	29	57
Others	7	14	Mixed org. unit	15	29
Not available	3	6	Not available	2	4
Sum	51	100	Sum	51	100

4.2 Results

The BIGSTEPS software 2.82 [32] has been used to obtain item calibrations. Important output statistics are the measure of difficulty (the logit values), the standard error, and the fit statistics of INFIT and OUTFIT for each item. Table 6 contains the results ordered by descending measure, i.e. difficult items have a higher measure than easy items. The fit values are all around 1 and therefore satisfying the expectation (see Table 3). The result is a flat list of ordered items. Although first conclusions can be drawn from such a list, the measures do not allow a division in distinct maturity levels. For clustering, we use the hierarchical cluster analysis of SPSS 19 (squared Euclidean distance, Ward's method) and set the desired number of maturity levels to five. The results of the cluster analysis are also exhibited in Table 6.

Table 6. Results of Rasch Algorithm

Logit	Error	Infit	Outfit	Item	Cl.	Short description
0.91	0.15	0.61	0.62	4.1.d	5	Proactive data quality management
0.79	0.14	1.26	1.23	1.3.c	5	Systematic and comprehensive measurement of real BI usage
0.72	0.14	1.2	1.18	1.3.b	5	Balanced Scorecard incl. quality, cost and user satisfaction
0.56	0.15	1.25	1.19	1.1.e	4	BI steering committee within business
0.56	0.14	1.02	1.01	2.2.d	4	Balanced mix of central and decentral organizational units
0.5	0.14	0.96	0.95	1.3.e	4	Portfolio mgt. for systematic BI roadmap
0.45	0.14	0.81	0.81	1.2.c	4	Regular update on BI strategy
0.39	0.14	0.59	0.61	1.3.d	4	Value-oriented dev. of BI, e.g. using business cases
0.37	0.13	0.76	0.84	2.3.b	4	Defined governance & standards for content
0.37	0.13	1	1.03	2.4.b	4	Dev. of BI based on standardized BI processes
0.37	0.13	0.62	0.62	2.6.b	4	Use of BI by middle-management
0.37	0.14	1.43	1.42	3.1.c.	4	Flexible, proactive analytics
0.33	0.13	1.06	1.05	1.2.b	4	BI strategy with focus on organization, processes as well as technology and tools
0.31	0.14	1.06	1.07	2.5.b	4	Central BI operations based on ITIL
0.24	0.14	0.85	0.86	4.1.a	3	Defined and documented roles for data quality mgt.
0.21	0.14	0.96	0.94	4.1.e	3	Std. definitions for key perf. indicators
0.2	0.14	1.13	1.14	4.2.e	3	BI operations based on well defined service-level-agreements (SLAs)
0.18	0.14	0.76	0.8	2.3.c	3	Defined governance & standards for management
0.17	0.14	0.47	0.48	4.2.d	3	Performance is satisfying for users
0.14	0.14	0.7	0.72	4.2.c	3	Timeliness: Usage of up-to-date tools and frontends
0.11	0.14	1.43	1.42	1.1.b	3	Multitude of decentral sponsors from business
0.11	0.14	0.91	0.9	4.1.b	3	Defined processes. for data quality mgt.
0.11	0.14	0.65	0.64	4.2.f	3	Cost efficient BI operations
0.09	0.13	1.29	1.36	2.1.d	3	Business partner - consulting of business lines
0.08	0.13	0.87	0.85	2.6.c	3	Use of BI by specialist
0.06	0.13	1.28	1.27	1.2.a	3	BI strategy with focus on techn. and tools
0.05	0.13	1.22	1.23	1.3.a	3	Std. cost and profit calculation of BI
0.02	0.14	0.82	0.83	1.1.d	3	Central influencing sponsor from business
0.01	0.13	1.09	1.08	3.1.d	3	Integration of different frontends, using drill-through from standard reports into OLAP cubes
0	0.14	0.81	0.8	4.2.b	3	Homogeneity: Usage of a few and coherent BI tools
-0.01	0.14	1.56	1.58	3.2.b	3	Partial integration in global systems (e.g. finance data warehouse)

Table 6. (*continued*)

-0.01	0.14	0.96	0.98	3.2.e	3	Balanced mix of central and decentral systems based on organizational structure
-0.03	0.13	1.25	1.24	2.6.d	3	Operational usage of BI
-0.06	0.14	0.91	0.91	1.1.f	3	BI steering committee within IT
-0.06	0.14	0.96	0.96	2.4.c	3	Development of BI using agile development methods (e.g. SCRUM)
-0.08	0.14	0.9	0.9	4.1.c	3	Most important data objects are defined for whole enterprise
-0.08	0.14	0.97	0.95	4.1.f	3	Standardized definitions for master data
-0.1	0.14	0.64	0.63	3.1.a	3	Static reports
-0.11	0.14	0.86	0.86	2.1.a	3	Defined governance & standards for dev.
-0.11	0.13	0.83	0.83	2.3.d	3	Operator of infrastructure
-0.15	0.15	1.02	1.02	2.4.d	2	Hybrid development of BI mixing agile development and waterfall methods
-0.22	0.14	1.2	1.21	2.1.c	2	Provision of standardized services
-0.27	0.14	1.31	1.3	2.2.c	2	Central BI organization
-0.28	0.14	1.38	1.38	3.2.d	2	Highly central data warehouse
-0.28	0.14	0.97	0.96	4.2.a	2	High availability: No breakdowns, maintenance in well defined and short time slots
-0.3	0.14	1.24	1.23	2.2.b	2	Decentral BI org. with central CIO org.
-0.31	0.14	1.39	1.44	2.6.a	2	Use of BI by top-management
-0.32	0.14	0.73	0.73	2.3.a	2	Defined governance & standards for tools and applications
-0.33	0.14	0.76	0.75	2.3.e	2	Defined gov. & standards for operations
-0.39	0.14	0.91	0.93	2.4.a	2	Dev. of BI based on std. IT processes
-0.45	0.15	1.19	1.21	1.1.c	2	Central influencing sponsor from IT
-0.49	0.15	1.62	1.59	2.1.b	2	Project oriented development
-0.55	0.16	0.72	0.79	3.2.f	2	Decentral, but harmonized systems (e.g. standardized master data)
-0.58	0.15	1.16	1.13	3.1.b	2	Ad-hoc analyses (OLAP)
-0.65	0.15	1.2	1.12	2.5.a	2	Central operations of BI
-0.81	0.16	0.88	0.92	1.1.a	1	Multitude of decentral sponsors from IT
-0.89	0.16	1.15	1.14	2.2.a	1	Decentral BI org. and responsibilities
-0.91	0.17	0.98	1.08	3.2.c	1	Decentralized warehouses and central enterprise warehouse

Based on the results of the application of the Rasch algorithm and the cluster analysis, an initial MM can be derived. Fig. 3 presents a sketch of the MM, based on the meta structure of strategy, organization & processes, IT and quality of service as outlined in the questionnaire structure.

The results can be interpreted as follows: Level 1 is populated by capabilities reflecting an early stage of decentralized BI organization. Level 2 emphasizes stronger orientation towards centrally managed BI in terms of governance and organization. Furthermore, services and processes are much more standardized on level 2. This could be interpreted as a first wave of optimized BI operations. On level 3, there is a strong focus on governance, strategy, and development. In addition, the BI infrastructure is also more balanced between centralized and decentralized

		Level 1		Level 2		Level 3		Level 4		Level 5	
		Item	Description	Item	Description	Item	Description	Item	Description	Item	Description
1. Strategy	1.1 Governance	1.1.a	Multitude of decentral sponsors from IT	1.1.c	Central influencing sponsor from IT	1.1.b 1.1.d 1.1.f	Multitude of decentral sponsors from business Central influencing sponsor from business BI steering committee within IT	1.1.e	BI steering committee within business		
	1.2 Strategy					1.2.a	BI strategy with focus on technology and tools	1.2.b 1.2.c	BI strategy with focus on organization, processes as well as technology and tools Regular update on BI strategy		
	1.3 Controls					1.3.a	Standardized cost and profit calculation of BI	1.3.d 1.3.e	Value-oriented development of BI, e.g. using business cases Portfolio management for systematic BI roadmap	1.3.b 1.3.c	Balanced scorecard incl. quality, cost, and user satisfaction Systematic and comprehensive measurement of real BI usage
2. Processes & organization	2.1 Role			2.1.b 2.1.c	Project oriented development Provision of standardized services	2.1.a 2.1.d	Operator of infrastructure Business partner - consulting of business lines				
	2.2. Organization	2.2.a	Decentral BI organization and responsibilities	2.2.b 2.2.c	Decentral BI organization with central CIO organization Central BI organization			2.2.d	Balanced mix of central and decentral organizational units		
	2.3. Governance processes			2.3.a 2.3.e	Defined governance & standards for tools and applications Defined governance & standards for operations	2.3.c 2.3.d	Defined governance & standards for management Defined governance & standards for development	2.3.b	Defined governance & standards for content		
	2.4 Change			2.4.a	Development of BI based on standardized IT processes	2.4.c 2.4.d	Development of BI using agile development methods (e.g. SCRUM) Hybrid development of BI mixing agile development and waterfall methods	2.4.b	Development of BI based on standardized BI processes		
	2.5 Run			2.5.a	Central operations of BI			2.5.b	Central operation of BI based on ITIL		
	2.6 Usage			2.6.a	Use of BI by top-management	2.6.c 2.6.d	Use of BI by specialists Operational usage of BI	2.6.b	Use of BI by middle-management		
3. IT	3.1 Frontend			3.1.b	Ad-hoc analyses (OLAP)	3.1.a 3.1.d	Static reports Integration of different frontends, e.g. drill-through from standard reports into OLAP cubes	3.1.c	Flexible, proactive analytics		
	3.2 Backend	3.2.c	Decentralized warehouses and central enterprise warehouse	3.2. d 3.2.f	Highly central data warehouse Decentral, but harmonized systems (e.g. standardized master data)	3.2. b 3.2.e	Partial integration of information in global systems (e.g. finance data warehouse) Balanced mix of central and decentral systems based on organizational structure				
4. Quality of service	4.1 Data quality					4.1.a 4.1.c 4.1.b 4.1.f 4.1. e	Defined and documented roles for data quality management Most important data objects are defined for the whole enterprise Defined processes for data quality management Standardized definitions for master data Standardized definitions for key performance indicators			4.1.d	Proactive data quality management
	4.2. System quality			4.2.a	High availability: No breakdowns, maintenance in defined and short time slots	4.2. b 4.2.c 4.2.d 4.2.e 4.2.f	Homogeneity: Usage of few and coherent BI tools Timeliness: Usage of up-to-date tools and frontends Performance is satisfying for users BI operations based on well defined service-level-agreements (SLAs) Cost efficient BI operations				

Fig. 3. Algorithm based Maturity Model

operations. BI applications are used by a broader scope of users on this level. Moreover, data quality and system quality have improved. This indicates a widespread diffusion of BI development and BI usage throughout the company. Level 4 is dominated by restructuring, value orientation and further optimization of BI. With growing BI experience, companies consolidate and streamline their widespread BI activities. The capabilities on level 5 are focused on the proactive and fact-based management of ongoing BI operations.

4.3 Discussion

The application of the proposed methodology yields in an initial version of a BI MM. It clearly follows an S-curve or normal distribution, with few capabilities at the beginning, an increasing number of capabilities on levels 2 and 3, and fewer additional activities on levels 4 and 5. A comparable development has also been proposed within the data warehouse MM by Watson et al. [41] and the TDWI BI MM [42]. It is also a basic assumption of MM construction (see section. 2.1). Furthermore, not only the number of allocations, but also the interpretation from a content perspective leads to a consistent and reasonable initial version of a BI MM. However, it should be clearly noted that such an initial MM needs additional iterations as proposed by the overall design science research approach [14, 15] as also for MM construction [18]; e.g. by leveraging focus groups for level and item discussion. Finally, it can be concluded that in the context of this BI-related analysis the proposed method has been demonstrated to be a useful means for empirically grounded and therefore more rigorous MM construction.

5 Conclusion and Outlook

MMs have become an established means in the IS community to systematically document and guide the development of organizations using archetypal capability levels. Despite its popularity, the concept has been criticized as lacking a generally accepted design methodology and formal basis, leading to a widespread development of MMs of disputable quality. This paper presents a statistical, psychometric-driven approach for designing MMs based on the Rasch algorithm and cluster analysis. Compared to previous applications of the Rasch algorithm in this field [22, 28, 29], the proposed method takes the current situation as well as corresponding targets into account. Moreover, the arbitrary assignment of items to maturity levels is overcome by applying a clustering algorithm.

The applicability of the method is limited to mature domains as there is a relative large sample needed for identifying items and derive maturity levels. This is also accompanied with larger efforts compared to other MM construction methods. Even though the result is more rigorous, it could lead to counterintuitive results during the first iterations and therefore may be lacking acceptance compared to a top-down development. Future research should be directed towards integrating the method into a full MM design process, i.e. using the Rasch algorithm for holistic organizational assessment. The algorithm could also be used for situational MM design (e.g. by using specific sample sets) and MM evolution on the basis of iterative survey execution.

References

1. Paulk, M.C., Curtis, B., Chrissis, M.B., Weber, C.V.: Capability Maturity Model, Version 1.1. IEEE Software 10, 18–27 (1993)
2. Ahern, D.M., Clouse, A., Turner, R.: CMMI Distilled: A Practical Introduction to Integrated Process Improvement. Addison-Wesley, Boston (2003)
3. Hakes, C.: The corporate self assessment handbook. Chapman & Hall, London (1996)
4. Fraser, P., Moultrie, J., Gregory, M.: The Use of Maturity Models/Grids as a Tool in Assessing Product Development Capability. In: IEEE International Engineering Management Conference (IEMC 2002), Cambridge, UK (2002)
5. Crosby, P.B.: Quality Is Free - The Art of Making Quality Certain. McGraw-Hill, New York (1979)
6. Gibson, C.F., Nolan, R.L.: Managing the four stages of EDP growth. Harvard Business Review 52, 76–88 (1974)
7. Pries-Heje, J., Baskerville, R.L.: Management design theories. In: Pries-Heje, J., Venable, J., Bunker, D., Russo, N.L., DeGross, J.I. (eds.) IFIP WG. IFIP Advances in Information and Communication Technology, vol. 318, pp. 263–281. Springer, Heidelberg (2010)
8. Becker, J., Niehaves, B., Pöppelbuss, J., Simons, A.: Maturity Models in IS Research. In: 18th European Conference on Information Systems (ECIS 2010), Pretoria, South Africa (2010)
9. Mettler, T., Rohner, P., Winter, R.: Towards a Classification of Maturity Models in Information Systems. In: 6th Conference of the Italian Chapter of AIS, Costa Smeralda, Italy (2009)
10. Pfeffer, J., Sutton, R.: Knowing "What" To do is not enough: Turning knowledge into action. California Management Review 42, 83–108 (1999)
11. Lahrmann, G., Marx, F., Winter, R., Wortmann, F.: Business Intelligence Maturity Models: An Overview. In: VII Conference of the Italian Chapter of AIS (itAIS 2010), Naples, Italy (2010)
12. Biberoglu, E., Haddad, H.: A Survey of Industrial Experiences with CMM and the Teaching of CMM Practices. Journal of Computing Sciences in Colleges 18, 143–152 (2002)
13. Winter, R.: Design Science Research in Europe. European Journal of Information Systems 17, 470–475 (2008)
14. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A Design Science Research Methodology for Information Systems Research. JMIS 24, 45–77 (2007)
15. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28, 75–105 (2004)
16. Simpson, J.A., Weiner, E.S.C.: The Oxford English Dictionary. Oxford University Press, Oxford (1989)
17. Rosemann, M., De Bruin, T.: Towards a Business Process Management Maturity Model. In: 13th European Conference on Information Systems (ECIS 2005), Regensburg, Germany (2005)
18. Becker, J., Knackstedt, R., Pöppelbuß, J.: Developing Maturity Models for IT Management - A Procedure Model and its Application. Business & Information Systems Engineering 1, 213–222 (2009)
19. Mettler, T., Rohner, P.: Situational Maturity Models as Instrumental Artifacts for Organizational Design. In: 4th International Conference on Design Science Research in Information Systems and Technology, Philadelphia, USA (2009)

20. de Bruin, T., Rosemann, M., Freeze, R., Kulkarni, U.: Understanding the Main Phases of Developing a Maturity Assessment Model. In: 16th Australasian Conference on Information Systems (ACIS 2005), Sydney, Australia (2005)
21. van Steenbergen, M., Bos, R., Brinkkemper, S., van de Weerd, I., Bekkers, W.: The design of focus area maturity models. In: Winter, R., Zhao, J.L., Aier, S. (eds.) DESRIST 2010. LNCS, vol. 6105, pp. 317–332. Springer, Heidelberg (2010)
22. Mettler, T.: Supply Management im Krankenhaus: Konstruktion und Evaluation eines konfigurierbaren Reifegradmodells zur zielgerichteten Gestaltung. Sierke, Göttingen (2010)
23. Utterback, J.M., Abernathy, W.J.: A dynamic model of process and product innovation. Omega 3, 639–656 (1975)
24. Hayes, W., Zubrow, D.: Moving on up: data and experience doing CMM-based software process improvement (1995)
25. vom Brocke, J.: Design Principles for Reference Modeling - Reusing Information Models by Means of Aggregation, Specialization, Instantiation, and Analogy. In: Fettke, P., Loos, P. (eds.) Reference Modeling for Business Systems Analysis, pp. 47–75. Idea Group, Hershey (2007)
26. Maier, A.M., Moultrie, J., Clarkson, P.J.: Developing maturity grids for assessing organisational capabilities: Practitioner guidance. Engineering Design Centre, Engineering Department, University of Cambridge, Cambridge, UK (2009)
27. Järvinen, P.: On Reviewing of Results in Design Research. In: 15th European Conference on Information Systems (ECIS 2007), St. Gallen, Switzerland (2007)
28. Rönkkö, M., Järvi, A., Mäkelä, M.M.: Measuring and Comparing the Adoption of Software Practises in the Software Product Industry. In: Wang, Q., Pfahl, D., Raffo, D.M. (eds.) ICSP 2008. LNCS, vol. 5007, pp. 407–419. Springer, Heidelberg (2008)
29. Dekleva, S., Drehmer, D.: Measuring Software Engineering Evolution: A Rasch Calibration. Information Systems Research 8, 95–104 (1997)
30. Bond, T.G., Fox, C.M.: Applying the Rasch Model. Fundamental Measurement in the Human Sciences. Fundamental Measurement in the Human Sciences. Lawrence Erlbaum Associates Publishers, Mahwah (2007)
31. Rasch, G.: Probabilistic Models for Some Intelligence and Attainment Tests. University of Chicago Press, Chicago (1980)
32. Linacre, J.M., Wright, B.D.: Bigsteps Rasch Software, Version 2.82 (1998), `http://www.winsteps.com/bigsteps.htm`
33. Bühner, M.: Einführung in die Test- und Fragebogenkonstruktion. Pearson Studium, München (2004)
34. Fischer, G.H., Molenaar, I.W.: Rasch Models: Foundations, Recent Developments, and Applications. Springer, Heidelberg (1995)
35. Likert, R.: A Technique for the Measurement of Attitudes. Archives of Psychology 140, 1–55 (1932)
36. Samuelson, P.A.: Foundations of Economic Analysis, Enlarged Edition. Harvard University Press, Harvard (1983)
37. Gordon, A.D.: Hierarchical Classification. In: Arabie, P., Hubert, L.J., De Soete, G. (eds.) Clustering and Classification, pp. 65–121. World Scientific Publishing, River Edge (1996)
38. Härdle, W., Simar, L.: Applied Multivariate Statistical Analysis. Springer, Berlin (2003)
39. Winter, R., Fischer, R.: Essential Layers, Artifacts, and Dependencies of Enterprise Architecture. Journal of Enterprise Architecture 3, 7–18 (2007)

40. Paech, B., Kerkow, D.: Non-Functional Requirements Engineering - Quality is Essential. In: Requirements Engineering: Foundation for Software Quality, REFSQ, Essen, Riga, pp. 27–40 (2004)
41. Watson, H.J., Ariyachandra, T., Matyska, R.J.: Data warehousing stages of growth. Information Systems Management 18, 42–50 (2001)
42. TDWI: 2008 TDWI BI Benchmark Report - Organizational and Performance Metrics for BI Teams. TDWI, Renton (2008)

Pattern-Based Approach for Designing with Diagrammatic and Propositional Conceptual Models

Wolfgang Maass[1,2] and Sabine Janzen[2]

[1] Research Center for Intelligent Media (RCIM),
Hochschule Furtwangen, D-78120 Furtwangen, Germany
{firstname.lastname}@hs-furtwangen.de
[2] Institute of Technology Management (ITEM),
University of St. Gallen, 9000 St. Gallen, Switzerland
wolfgang.maass@unisg.ch

Abstract. A conceptual modeling approach for Ubiquitous Information Systems (UIS) is presented as a central part of a UIS design methodology. Three conceptual models are used for step-wise derivation of machine-executable design models for distributed service infrastructures: narratives, pattern-based diagrammatic conceptual models (Pre-Artifacts), and formalized propositional conceptual models.

Keywords: Conceptual modeling, design methodology, Ubiquitous Information Systems, semantic technologies, patterns.

1 Introduction

The class of Ubiquitous Information Systems (UIS) has recently gained attention [1, 2] but is governed by ad-hoc methods, e.g. “wild-west” prototyping. UIS require design approaches that keep a holistic view of situations in which single users and groups interact with one another and with accessible services. In the following, a design methodology for UIS is presented that is centered around three types of conceptual models and corresponding translation procedures.

Design teams for Information Systems are heterogeneous, with members from different fields such as domain experts, various users, decision makers, IT architects, analysts, developers, and Marketing experts. All ideas, expertise, experience, and expectations of these members are brought together for building a homogenous understanding of a future Information System. Explication and communication means are required for building these understandings on various levels. Non-technical members intend to build an Information System that supports their business, social, and communication needs while technical members focus more on engineering aspects of the technical realization of the system. Shared understandings of design teams are described by various conceptual models (CM) that are used during design phases of Information System development [3]. Central to conceptual modeling is the identification of important concepts and relations [3, 4] semantically described by shared vocabularies [5]. Shared vocabularies are either implicitly defined as being part of a mutual understanding in a community or explicitly defined in forms of

H. Jain, A.P. Sinha, and P. Vitharana (Eds.): DESRIST 2011, LNCS 6629, pp. 192–206, 2011.

machine-processable representations [6]. In the latter case, the logic of a CM can be evaluated and matched with other CMs which is important for re-use [7]. CMs abstract from technical issues and focus on aspects of situations in which users and user groups perform activities that are supported by information and communication services [3]. A CM is represented by a conceptual modeling language (CML), such as Entity-Relationship [8] models or the Unified Modeling Language (UML) [9]. From an IS development process perspective, CMs are used during analysis, design, and realization phases [3].

Ubiquitous Information Systems (UIS) provide means for supporting single actors and groups in real-world situations by services over ubiquitous computing technologies anywhere and anytime [2, 10]. Little research has been done so far on dedicated design methodologies and conceptual modeling for UIS [11, 12]. Klemmer and Linday investigate CMs for designing tangible user interfaces based on physical input devices [11]. Janzen et al. propose a design methodology for UIS that models situations by narratives and semi-structured representations [12]. In this article, we discuss a conceptual modeling approach based on three aligned types of conceptual models that (1) supports holistic and explicit representations of communication and collaboration situations for UIS and (2) uses service infrastructures as a means for supporting social functions. Next, the role of conceptual models is discussed for Information Systems and UIS in particular. This sets the scene for our design methodology for UIS followed by a detailed discussion of a conceptual modeling approach based on diagrammatic and formal propositional CMs. Results and future work close this article.

2 Conceptual Modeling

Conceptual modeling is a key topic for design science theories [3]. Several CML are proposed with a focus on (1) business process modeling (e.g., [13]), (2) general software engineering (e.g., [9]), (3) semantic data models (e.g., [8]), and meta-data models and computational ontologies [14]. Grammars provided by CMLs require ontologies for defining the fundamental entities and structures that shall be focused by CMs [3].

In an idealized form, conceptual modeling transforms existing explicit CMs or implicit mental models of members of a design team into integrated CMs, *CM(D, L, O)*, by means of a modeling method *M* and a conceptual language *L* based on a domain ontology *D* and a fundamental information systems ontology *O* (Fig. 1). Hence, CMs are a type of shared mental model that support mappings from application domains to CMs and from CMs to views of an information system. Useful conceptual modeling approaches “should enable both mappings without loss of information” [3]. The distinction between CMs and design models for information systems gets blurred if CMs can be executed [3] as intended by CMs based on formal ontologies [15].

In different phases, IS design teams use different types of CML. Consistency, syntactic, and semantic interoperability are major obstacles for working with different CMLs. For instance, UML and the Unified Software Development Process (USWDP) provides a set of CMLs for representing different conceptual aspects and a qualitative

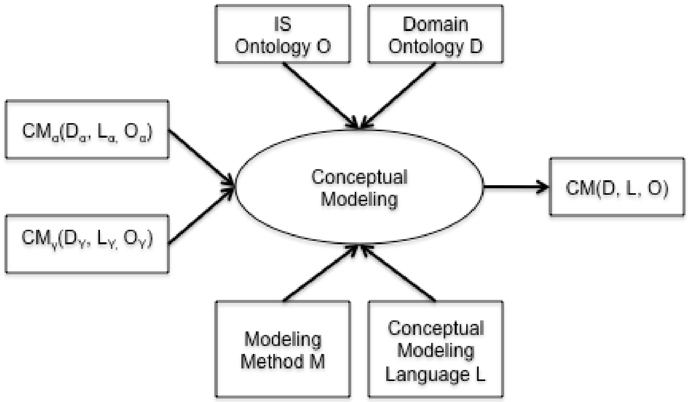


Fig. 1. Generic model of conceptual modeling

procedure for iterative and incremental software development [9]. But UML and USWDP do not give clear guidance for building integrated CMs caused by cognitive misdirection, semantic inconsistency, inadequacy, and ambiguity of modeling concepts [16]. Some CML of the UML are qualitative, such as use cases, while others are formal, such as state transition diagrams. For instance, the Rational Unified Process (RUP) provides 159 key resulting artifacts that are created and used during the software development process that are managed by at least four systems [17]. Use case centered development focuses on single observable results of value for a particular actor [9, 18]. For fixing this atomistic view, the concept of a *summary use case* was introduced that textually describes how various instances of a use-case combine to achieving an overarching goal [19]. No structure and guidance for writing summary use cases are given. Thus, use case modeling conveys localized perspectives, which often atomizes overall understandings of a target system [16].

Conceptual modeling frameworks for UIS have to cope with more complex requirements than the more strongly constrained Information Systems for office settings [1]. Contents shall be seamlessly provided by any kind of mobile or embedded device based on loosely coupled service infrastructures while users are moving in physical environments [1]. Hence, situated communication and collaboration of user groups in physical environments are far more complex than well-structured online environments, highly dynamic, and context-dependent on various dimensions [20]. This requires that contents can effortlessly move over loosely coupled and distributed service infrastructures, for instance, supported by semantically annotated contents. Next, a design methodology for UIS is briefly described before the underlying conceptual modeling approach is presented.

3 Design Methodology for Ubiquitous Environments

Designing UIS does not exclusively depend on technical issues but also on aspects concerning, for instance, users, social interactions, and physical surroundings. Environments of UIS cannot be fully specified, i.e. UIS designs should be flexible enough to cope with a range of unpredictable events and entities. This is in contrast with fully specified, artificial digital environments of traditional IS. A basic hypothesis for the design of UIS is that this general requirement for flexibility can be supported by strongly modularized computing environments and dedicated design principles for composing computational modules [2]. Thus, the following limitations of design methods for purely digital IS (e.g., [4, 9, 21]) should be overcome by a design methodology for UIS: (1) consideration of physical objects (e.g., [22]) and (2) contextualized computational modules (e.g., [20]). Contextualized computational modules describe logically coherent interactions not only by its functionality but also with respect to requirements on contents, social organization, interactions, and supporting services. Previous design science research identified seven development principles for the design of information systems, which should be addressed by a design method [23, 24]. Based on these principles, we derived a design methodology for UIS, called *Content-Centered Design of Ambient Environments (CoDesA)* [12]. CoDesA consists of four phases: (1) Identification of Problem & Needs, (2) Design of Solution, (3) Development of Solution and (4) Evaluation of Solution. These phases consist of nine tasks: (1) Identification of problem and needs, (2) Derivation of situations (narratives), (3) Derivation of diagrammatic CM (Pre-Artifacts), (4) Evaluation of Pre-Artifacts, (5) Derivation of formal propositional CMs, (6) Formalization of system design, (7) Implementation of formalized system design, (8) Evaluation of solution, and (9) Product development. CoDesA was tested in various UIS development projects. In the following, definition of diagrammatic CM and derivation of formal propositional CM are focused (task 3 and 5). In particular it is presented how diagrammatic conceptual patterns help to solve problems with ambiguities of qualitative CMs (task 3) and how formal web-based CML can be used for deriving executable CMs (task 5).

4 Conceptual Modeling of UIS

4.1 Fundamental Information System Ontology AISM

Conceptual models are strongly influenced by basic conceptualizations of Information Systems [3] (Fig. 2). In general, information systems are compounds of social systems, information, and service systems that use information technology infrastructures for realization of desired situations [25-27]. With the Abstract Information System Model (AISM), we bring together these three conceptual classes for conceptual models of Information Systems with the additional dimension of physical entities that is required for UIS (cf. Fig. 2):

1. Social system: the set of roles available with a set of attributes, such as rights, obligations, and prohibitions, and actions performed by role-taking actors
2. Information sphere: all information objects used within the realm of an IS
3. Physical object system: the set of physical entities available within all situations in which a UIS can be used
4. Service system: the set of all digital and physical services available within all situations in which an IS can be used

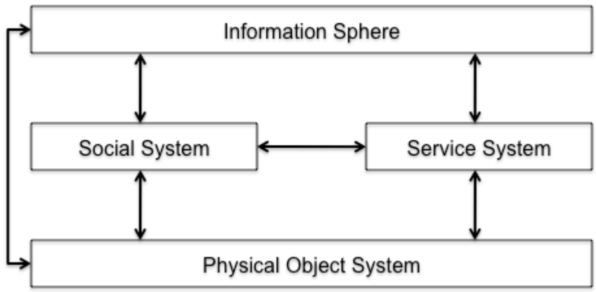


Fig. 2. Abstract Information System Model (AISM)

Information objects that are used in situations by the social system of role-taking actors are defined within an information sphere based on supporting services. Actors use roles, information objects, and services for implementing situations in work and other contexts. An organization consists of structural elements, in particular role systems that describe the capabilities of roles and attributes, and dynamic structures based on interactions that describe procedural aspects of an organization. Interactions are explications of task requirements that are described by directed relations between roles. Communication, as a sub-class of interactions, transfers information that, in turn, refers to information objects. Information objects are abstract conceptions of any kind of content, such as speech, written language, graphics, or digital contents. Information objects generally require support given by a role-taking actor or a service. In the other case, external services can create information objects and make them available to other services or role-taking actors. Services provide functional capabilities to roles and other services. Services that support roles are called interface services. Interface services provide graphical, tangible, speech or other interfaces by which role-taking actors access services for achieving some situation-specific goals. Mobile, Pervasive, and Ubiquitous computing [10, 28-30] are approaches by which Information Systems (a) extend from spatially restricted access to information spheres to temporally and spatially unrestricted access forms and (b) invisible embedding of information technologies in physical environments. The physical object system encompasses all physical objects that are relevant for the design model of an Information System. In the following, AISM takes the role of the IS ontology O (Fig. 1).

4.2 Pre-Artifacts

Situations are instance-based descriptions of interactions between entities in an environment that use concepts and relations defined by shared vocabularies. CMs described by a CML capture situations and frame design discussions [31]. Our methodological approach is based on three CM types: narrative CM, diagrammatic CM (Pre-Artifacts), and propositional CM. In task 2, narratives describe identified situations. The mode of narrative thinking is highly context-sensitive, anchored in situations, articulated in temporal sequences, around individual and group intentions and actions [32]. Narratives are effective means for building and understanding situations of future Information Systems because of their capacity to provide discourse information and sequential orderings of interactions between actors [33, 34]. Next narratives are translated into diagrammatic CM, called Pre-Artifacts (task 3). Based on the AISM, Pre-Artifacts conceive usage situations by emphasizing requirements on social structure, information objects, physical objects, and services in a coherent structured manner. All core entities are identified in narratives that fit to these conceptual categories [12].

Similarly, relations are extracted that connect these entities. Analogue to use cases, Pre-Artifact are described on instance level but are used as prototypes for class descriptions [3]. The concept of a Pre-Artifact resembles the basic concept of use cases because it also describes logically consistent parts of a situation. But Pre-Artifacts are structured by an underlying IS model (AISM) with a set of defined concepts (information object, role, services, and interactions) while use cases are neutral with respect to IS models. Another differentiating factor is that the Pre-Artifact model is geared towards role-based designs that explicitly demand relationships between information objects and roles.

For heterogeneous information spheres this is important for qualification of information with respect to, for instance, reputation, responsibility, and copyrights and also for explanation-based systems. For these kind of typical IS requirements, use cases are not specific enough and require external guidelines that, in turn, increases the complexity for their application in design situations. In the following, we focus on modeling with diagrammatic CM patterns, called Pre-Artifact Patterns, and translation of resulting diagrammatic CM into formal propositional CM.

4.3 Pre-Artifact Patterns

Analysis of Pre-Artifacts in several UIS development projects showed re-occurring structures similar to the notion of design patterns as used in architecture [35] and Software Engineering [36]. They represent means for reusable CMs for Information Systems. We identified seven Pre-Artifact patterns (cf. Fig. 3) that are elaborated in the following.

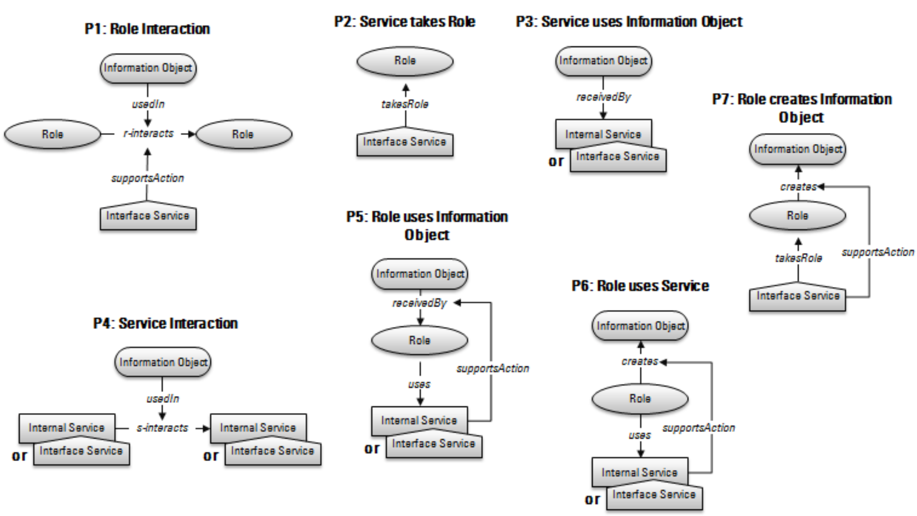


Fig. 3. Pre-Artifact patterns

Role Interaction Pattern (P1): This pattern describes a situation in which two or more role-taking actors interact with one another by exchanging information objects supported by an interface service, e.g., mail communication between sender and receiver. The interaction between roles is described by a generic property called *r-interacts*. The interface service is only used as a communication channel.

Service takes Role Pattern (P2): This pattern represents a situation in which a role is taken by an interface service. For instance, Wikipedia provides information and takes a role with connotated social attributes, such as reputation and credibility.

Service uses Information Objects Pattern (P3): expresses that an internal or interface service receives information objects without human interventions. This is a simplification of the Service Interaction pattern. It is used when a providing service is not important for a CM. For instance, stock information used by a local service and received from a cloud infrastructure.

Service Interaction Pattern (P4): This pattern describes the interaction relationship of two interface or internal services with no interaction with human actors. Within this interaction that is represented by *s-interacts,* an information object is used. The interaction relationship between services is described by *s-interacts* while roles are connected by *r-interacts* as mentioned before. For instance, a local temperature service sends data to a central weather service. In contrast to the Role Creates Information Object pattern, this pattern supports system designs that do not use role-based on service level.

Role uses Information Object Pattern (P5): In situations with direct manipulation of information objects, this pattern allows to express that a role receives an information object by using an internal or interface service. This means a role-taking actor can actively receive an information object supported by a service. For instance, a CEO who uses a business intelligence service for accessing corporate sales information.

Role uses Service Pattern (P6): This pattern describes a situation with a role-taking actor creating an information object. Therefore, the actor uses a service that supports the creation of an information object, e.g., a nurse who creates a status report for a patient by a healthcare reporting service.

Role creates Information Object Pattern (P7): By this pattern a service creates an information object by taking a role which links an information object to a service. This pattern supports role-based system designs. For instance, a vital sign monitoring system can take a role that allows it to create emergency alerts. Created alerts are directly linked with this service via a role.

4.4 Method for Conceptual Modeling with Pre-Artifact Patterns

Conceptual modeling of Pre-Artifacts is improved by Pre-Artifacts patterns because they provide conceptual structures as basic building blocks for IS designs. The construction of Pre-Artifacts is guided by a method with five steps based on instantiation and integration of Pre-Artifact patterns. Each step pursues a sub goal for constructing the Pre-Artifact step by step and proclaims specific Pre-Artifact patterns that help to achieve the objectives (cf. Tab. 1).

Table 1. Appliance of Pre-Artifact patterns within steps of defining Pre-Artifacts

Patterns P / Steps	Step 1	Step 2	Step 3	Step 4	Step 5
P1: Role Interaction	-	x	-	-	-
P2: Service takes Role	-	-	x	-	-
P3: Service uses Information Object	-	-	-	x	-
P4: Service Interaction	-	-	-	x	-
P5: Role uses Information Object	-	-	-	-	x
P6: Role uses Service	-	-	-	-	x
P7: Role creates Information Object	-	-	x	-	-

Step 1: Definition of *Information Objects* in *Infosphere*. All information objects that occur in a narrative are defined as Information Objects (IO) in the Infosphere.

Step 2: Definition of user-system or user-user interactions related to *Information Objects*. Within this step, interactions between users or user and system related to newly generated information objects have to be defined. These interactions take place between *Roles* in the *Social System* exclusively. Thus step 2 connects infosphere and social system. Interactions between user and system are always supported by a service of the *Service System* that is defined in Step 3. The requirements of this step are fulfilled by the application of the *Role Interaction pattern* exclusively.

Step 3: Definition of *Roles* taken by *Services*. Next, an interface service has to be defined that takes a role for creating the new information object that will be used in the interaction. Therefore, the service has to take a role in the interaction. Either a service is linked to a role that was already defined in step 2 or it adds a new role. Thus step 3 connects infosphere and social system with the service system. To manage this step, the *Role Creates Information Object pattern* is applied to define the creation of the information object by a role taken by a service.

Step 4: Definition of supporting *Internal Services*. To create new information objects, generic information sources are needed as mentioned before. The interface service that supports the creation of a new IO needs access to these sources. Therefore, *Internal Services* for all remaining information objects in the Infosphere are specified. The interaction between services regarding the information objects is realized by applying the *Service Interaction pattern*.

Step 5: Definition of user initiative. If a user role initiates an interaction with the system that means using the system in a proactive way, this situation is modeled by using the *Role uses Service* or *Role uses Information Object* pattern. The role uses a service to create or receive an information object, for instance, the user wants to leave a message for another user. This action is indirectly supported by a service.

5 Example

Next, an example will be given for deriving a diagrammatic CM from a narrative CM based on the Pre-Artifact CML, Pre-Artifact patterns, and CM method. Finally, three approaches of translating Pre-Artifacts into propositional CMs are discussed and exemplified.

5.1 Defining Pre-Artifacts

Step 1: Definition of *Information Objects* in *Infosphere*. Fig. 4 shows this modeling step by means of an exemplary Pre-Artifact that shall represent the narrative: "*It's Thursday morning. I get site-specific weather information when I am brushing my teeth in the bathroom.*" The figure shows that the modeling person has specified the goal "*Getting weather information for user's location*" that is assigned to the user in the situation. Furthermore, the information object *site-specific weather information* is defined. This information object has to be created in the situation based on the required information objects *global weather information* and *location*.

Step 2: Definition of user-system or user-user interactions related to *Information Objects*. In the exemplary Pre-Artifact (cf. Fig. 4) an interaction between a *Personalized Weather Assistant* and the *User* was modeled that is supported by a *Personalized Weather Service*. Subject of the interaction is the IO *site-specific weather information*.

Step 3: Definition of *Roles* taken by *Services*. In our example, the Personalized Weather Service takes the role of the Personalized Weather Assistant that creates the IO site-specific weather information. The interface service supports this action indirectly (cf. Fig. 4). To express the plain role-taking by a service without a creating function, the *Service takes Role* pattern can be applied.

Step 4: Definition of supporting *Internal Services*. The exemplary Pre-Artifact (cf. Fig. 4) shows the definition of two internal services *Weather Service* and *User Context Service* that feed a *Personalized Weather Service* with global weather information and location data.

Step 5: Definition of user initiative. Step 5 is not required in this example because the user does not initiate interactions with the UIS (cf. Fig. 4).

Next, Pre-Artifacts are translated into propositional CMs (task 5). Currently this translation is a manual task but we work on an automatic translation mechanism so that designers are not required to deal with formal logics. The objective of this translation is the creation of specifications for later system designs [3] as well as machine-processable CMs that can be verified [14]. Considering the method of formalization, there are several opportunities, for instance Unified Modeling Language (UML), entity-relationship model (ER) or a formalization, for instance, based on ontologies by means of RDF (http://www.w3.org/RDF) or OWL (http://www.w3.org/TR/owl-ref). The use of computational ontologies for conceptual modeling by means of a pattern-based approach has already been investigated [37, 38]. Bera et al. (2010) identified some unique features of OWL that are not available in ER model and in UML. Amongst others, OWL is implementable, which means OWL ontologies are machine-readable, and thus computational. Furthermore, OWL constructs are independent, i.e. classes can exist independent of instances or properties and properties are independent of classes. Concerning the verification aspect, OWL allows inferences and automated reasoning support. Beside these advantageous features of OWL, there are also difficulties in using OWL for the formalization of Pre-Artifacts. Bera et al. (2010) determine that there are no clear rules how to map from domain information as represented by Pre-Artifacts to OWL constructs similar to the intended propositional CMs.

There are at least three approaches of translating Pre-Artifacts into propositional CMs. Each option was tested by modeling three exemplary Pre-Artifact patterns (Role Interaction, Role creates Information Object, Service Interaction) that were used for generating the propositional CM in Section 5.1.

5.2 Translation of Pre-Artifacts into Propositional Conceptual Models

Approach 1. This option was realized by representing each Pre-Artifact pattern by a unique formal propositional model. The approach leads to redundant concepts when integrating propositional models into a complex propositional CM. A modeling person has to specify equivalences to resolve these redundancies; e.g., in our example (Fig. 4) the concept type “Role” occurs three times because three of the imported patterns contain this concept type. This procedure demonstrates the aforementioned lack of modeling guidelines and constraints [14].

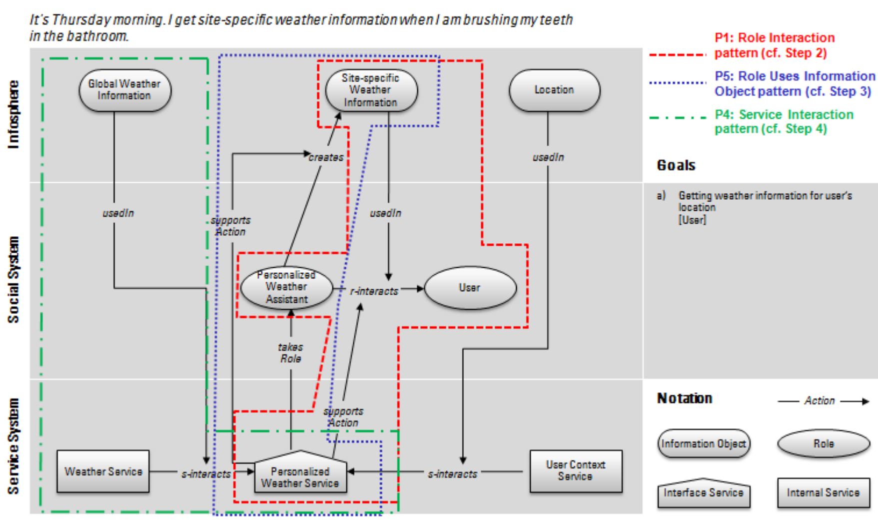


Fig. 4. Translation of a narrative in a Pre-Artifact based on three Pre-Artifact patterns

Approach 2. Following the approach by Bera et al. (2010) of using a “philosophical ontology” to derive guidelines on how OWL constructs can be applied in the modeling of propositional conceptual models, a Pre-Artifact Model is created that represents a “vocabulary” and generic object properties of Pre-Artifact patterns. The Pre-Artifact Model consists of 12 concept types and 8 generic object properties. It represents basic entities of AISM: *InformationObject, Role* and *Service* with sub-classes *Interface* and *Internal Service*. Furthermore, a super-class *Action* is defined that contains further sub-classes that specify diverse types of pattern actions: *Creation, Receiving* and *Interaction* with sub classes *R_Interaction* and *S_Interaction*. The decision to model most of the pattern relations by means of additive concepts is due to the fact that these relations represent three-way connections. A second opportunity would be to use property chains in OWL 2 that support transitive relationships between objects [39]. The advantage of the former opportunity lies in adding actions as specific concepts to the social system. This allows a differentiated consideration and extensibility by further properties. Furthermore, the model consists of 8 generic object properties: *initiatesInteraction, finalizesInteraction, initiatesAction, isResultOfAction, supportsAction, takesRole, usedIn* and *usesService*. Each pattern ontology imports the Pre-Artifact Model. Afterwards, the pattern ontologies specify relevant generic object properties with additional concepts. When integrating the pattern ontologies in the propositional CM, ambiguous assignments of object properties to specific patterns occur. Because of lack of clear results and statements, the second approach is not a proper solution for handling the lack of modeling guidelines.

Approach 3. In this approach, the notion of the Pre-Artifact Model as well as the integration of this "vocabulary" into pattern ontologies is adopted. But, for the specification of pattern-specific object properties based on the generic properties of the model, inheritance structures of object properties are used. That means each pattern defines sub properties of the relevant object properties imported from the model. Therefore, super-properties and concepts of the Pre-Artifact Model remain unchanged. In this context, the OWL feature is used, that OWL constructs are independent, i.e. properties can exist independent of classes [14]. Based on this approach, clear assignments of specified object properties to specific patterns are realized. Conceptual modelers will be supported by modeling guidelines because of a canalization of modeling options. The propositional conceptual model can be modeled in an incremental way by importing patterns step by step according to the requirements of the Pre-Artifact.

5.3 Example

Considering CoDesA Task 5 – the formalization of propositional Pre-Artifacts -, the aforementioned third approach was applied. To model the propositional Pre-Artifact based on the diagrammatic Pre-Artifact, the expressiveness of the semantic pool of Pre-Artifact patterns is used. This approach is based on the Pre-Artifact Model, the pool of Pre-Artifact Patterns, and propositional CMs. For derivation of propositional CM tools for modeling formal web-based representations are required, e.g., Protégé (http://protege.stanford.edu/). After generating an empty OWL file in Protégé, required Pre-Artifact patterns are imported by their URL. According to the procedure of defining Pre-Artifacts (cf. Section 5.1), the formalized model of the pattern *RoleInteraction* is imported. Then, the relevant concepts of the pattern are instantiated, e.g., by creating an instance of the concept "Role" named "User". To represent the interaction between *User* and *PersonalizedWeatherAssistant*, an instance of the concept *R-Interaction* is created. For linking both roles with the instance of R-Interaction, the formalized pattern offers the specified object properties *initatesR_Interaction* and *finalizesR_Interaction* that inherit from the super-properties *initiatesInteraction* and *finalizesInteraction*. Within the proceeding formalization, the formalized patterns *Role uses IO* and *Service Interaction* are imported. Note, each pattern automatically imports the Pre-Artifact Model. The result of the formalization is an OWL description that represents the exemplary narrative in a formal and computational way.

6 Discussion

Explication and integration of individual understandings of different members of a design team are central tasks of conceptual modeling for Information Systems. CMLs with an origin in Computer Science are technical languages that are difficult to use by non-technical design team members [16]. Therefore additional CMLs are required that support capturing different aspects of CMs. We presented a design methodology with three types of CMs for translating individual mental models, into narratives, diagrammatic CMs (Pre-Artifacts), and finally formalized propositional CMs. Thus shared understandings are incrementally supported from qualitative, textual descriptions of complete situations into, semi-structured representations with consolidated conceptual structures, and machine-processible, propositional representations based on formal ontologies. We have already evaluated by initial empirical studies that narratives and Pre-Artifacts are useful tools for modeling complex Information Systems. The Pre-Artifact modeling task is supported by a pattern-based approach that provides generic conceptual modules. Translation of Pre-Artifacts into formalized propositional CM support IT experts in rapid prototyping of Information Systems based on distributed service infrastructures, as common for UIS.

We have applied this modeling approach to a real world situation. Resulting formal propositional CMs are directly used as design models that can be executed on semantic technology infrastructures [40]. Thus, the three-step conceptual modeling approach of CoDesA supports rapid prototyping for complex UIS by

1. keeping the holistic structure of situations that supports an integrated understanding of interactions within complex socio-technical systems
2. providing narratives as qualitative CMs that support integrated conceptual views of design members from various domains,
3. structured pattern-based translation of narratives into Pre-Artifacts, and
4. algorithmic translation of Pre-Artifacts into formal propositional CM that can be executed on semantic technology infrastructures.

This research is a key step towards a structured conceptual modeling process in particular for complex UIS leveraging distributed service infrastructures in the sense of a utility computing model [41]. Whether CoDesA and Pre-Artifacts are sufficient means for modeling CM for UIS has to be proven in further projects. Furthermore it is an open issue whether OWL is sufficient as a formal language for automatic translation of diagrammatic CM into propositional CM of complex UIS.

7 Conclusion and Future Work

A three-step conceptual modeling approach was presented as an integral part of a design methodology for UIS. It was argued that a pattern-based modeling approach provide structure and guide conceptual modelers without loss of too much freedom.

We have shown the importance of IS ontologies and how they fundamentally affect conceptual models. The model of conceptual models provides a concise roadmap for future research. For instance, mapping of narratives derived by different modelers as input for a Pre-Artifact modeling task. Currently we work on empirical evaluations of the effectiveness of our conceptual modeling approach.

Acknowledgements

This paper is based on work done in the project 'Interactive Knowledge Stack (IKS)' (FP7 231527) co-funded by the European Commission.

References

1. Lyytinen, K., Yoo, Y.: Research Commentary: The Next Wave of Nomadic Computing. Information Systems Research 13(4), 377–388 (2002)
2. Yoo, Y.: Computing in Everyday Life: A Call for Research on Experiential Computing. Mis Quarterly 34(2), 213–231 (2010)
3. Wand, Y., et al.: Theoretical foundations for conceptual modelling in information systems development. Decision Support Systems 15, 285–304 (1995)
4. Markus, M.L., Majchrzak, A., Gasser, L.: A design theory for systems that support emergent knowledge processes. MIS Quarterly 26(3), 179–203 (2002)
5. March, S., Smith, G.: Design and natural science research on information technology. Decision Support Systems 15, 251–266 (1995)
6. Chandrasekaran, B.: Design Problem Solving: A Task Analysis. AI Magazine 11(4), 59–71 (1990)
7. Purao, S., Storey, V.C., Han, T.D.: Improving Reuse-Based System Design with Learning. Information Systems Research 14(3), 269–290 (2003)
8. Chen, P.: The Entity-Relationship Model–Toward a Unified View of Data. ACM Transactions on Database Systems 1(1), 9–36 (1976)
9. Booch, G., Rambaugh, J., et al.: The Unified Modeling Language User Guide. Addision-Wesley, Redwood City (1999)
10. Lyytinen, K., Yoo, Y.: Issues and Challenges in Ubiquitous Computing. Communication of the ACM 45, 62–65 (2002)
11. Klemmer, S., Landay, J.: Toolkit Support for Integrating Physical and Digital Interactions. Human-Computer Interaction 24, 315–366 (2009)
12. Janzen, S., Kowatsch, T., Maass, W.: A Methodology for Content-Centered Design of Ambient Environments. In: Winter, R., Zhao, J., Aier, S. (eds.) DESRIST 2010. LNCS, vol. 6105, pp. 210–225. Springer, Heidelberg (2010)
13. Scheer, A.-W.: ARIS. Modellierungsmethoden, Metamodelle, Anwendungen, 3rd edn. Springer, Berlin (1998)
14. Bera, P., Krasnoperova, A., Wand, Y.: Using Ontology Languages for Conceptual Modeling. Journal of Database Management 21(1), 1–28 (2010)
15. Evermann, J.: A UML and OWL description of Bunge’s upper-level ontology model. Software and Systems Modeling 8(2), 235–249 (2009)
16. Simons, A., Graham, I.: 30 Things that Go Wrong in Object Modeliling with UML 1.3. In: Kilov, H., Rumpe, B., Simmonds, I. (eds.) Behavioral Specifications of Businesses and Systems, pp. 237–257. Kluwer Academic Publishers, Dordrecht (1999)

17. Kruchten, P.: The Rational Unified Process-An Introduction, 3rd edn. Addison-Wesley, Reading (2003)
18. Jacobson, I., Griss, M., Jonsson, P.: Software Reuse: Architecture, Process and Organisation for Business Success. Addison-Wesley and ACM Press, Reading MA (1997)
19. Cockburn, A.: Writing effective use cases. Addison-Wesley, Upper Saddle River (2001)
20. Dey, A., Abowd, G.: Towards a better understanding of context and context-awareness. College of Computing, Georgia Institute of Technology (1999)
21. Jackson, E.B., Jackson, R.L.: Industrial information systems. Academic Press, London (1978)
22. Ross, P., Keyson, D.: The case of sculpting atmospheres: towards design principles for expressive tangible interaction in control of ambient systems. Personal Ubiquitous Computing 11(2), 69–79 (2007)
23. Markus, M.L., Keil, M.: If We Build It, They Will Come: Designing Information Systems That People Want to Use. Sloan Management Review 35(4) (1994)
24. Pries-Heje, J., Baskerville, R.L.: The Design Theory Nexus. Mis Quarterly 32(4), 731–755 (2008)
25. Lechner, U., Schmid, B.F.: Communities and Media – Towards a Reconstruction of Communities on Media. In: 34th Annual Hawaii International Conference on Systems Sciences (HICSS-34) (2001)
26. Orlikowski, W.J., Barley, S.R.: Technology and Institutions: What can Research on Information Technology and Research on Organizations Learn from each others? MIS Quarterly 25(2), 145–165 (2001)
27. Lamb, R., Kling, R.: Reconceptualizing Users as Social Actors in Information Systems Research. MIS Quarterly 27(2), 197–235 (2003)
28. Weiser, M.: The computer for the 21st century. Pervasive Computing 1(1), 18–25 (2002); Scientific Amercian (1991)
29. Ngai, E.W.T., et al.: Mobile commerce integrated with RFID technology in a container depot. Decision Support Systems 43(1), 62–76 (2007)
30. Varshney, U., Vetter, R.J., Kalakota, R.: Mobile commerce: a new frontier. Computer 33(10), 32–38 (2000)
31. Hult, L., Irestig, M., Lundberg, J.: Design Perspectives. Human-Computer Interaction 21, 5–48 (2006)
32. Zukier, H.: The Paradigmatic and Narrative Modes in Goal-Guided Inference. In: Sorrentino, R., Higgins, E. (eds.) Handbook of Motivation and Cognition: Foundations of Social Behavior, pp. 465–502. Guilford Press, New York (1986)
33. Hinchman, L., Hinchman, S. (eds.): Memory, identity, community: the idea of narrative in the human sciences. State University of New York Press, Albany (1997)
34. Kuechler, W., Vaishnavi, V.: On theory development in design science research: anatomy of a research project. European Journal of Information Systems 17, 489–504 (2008)
35. Alexander, C.: A Pattern Language: Towns, Buildings, Construction. Oxford University Press, Oxford (1977)
36. Gamma, E., et al.: Design Patterns. Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading (1994)
37. Clark, P., Thomspon, J., Porter, B.: Knowledge Patterns. In: Proc. of KR 2000. Morgan Kaufmann, San Francisco (2000)
38. Gangemi, A.: Ontology design patterns for semantic web content. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 262–276. Springer, Heidelberg (2005)

39. Motik, B., Patel-Schneider, P., Cuenca Grau, B.: OWL 2 Web Ontology Language: Direct Semantics (April 22, 2010), http://www.w3c.org/2007/OWL/wiki/Direct_Semantics
40. Carroll, J.J., et al.: Jena: implementing the semantic web recommendations. In: Proceedings of the 13th International World Wide Web Conference on Alternate Track papers and Posters, pp. 74–83. ACM, New York (2004)
41. Carr, N.: The End of Corporate Computing. MIT Sloan Management Review 46(3), 67–73 (2005)

Enacted Software Development Routines Based on Waterfall and Agile Software Methods: Socio-Technical Event Sequence Study*

B. Veeresh Thummadi[1], Omri Shiv[1], Nicholas Berente[2], and Kalle Lyytinen[1]

[1] Case Western Reserve University, Cleveland, Ohio USA
{Veeresh.Thummadi,Omri.Shiv,Kalle.Lyytinen}@case.edu
[2] The University Of Georgia, Athens, Georgia USA
Berente@uga.edu

Abstract. In recent decades, "agile" software development methodologies have been put forth as an alternative to traditional "waterfall" methodologies. These agile methods advance a fundamentally different approach to software development. Empirical evidence indicates differences between the two with respect to outcomes and development experience. Yet little is known to what extent the actual development practices based on either agile or traditional life cycle methodologies differ. In the current study we examine the variation in performative routines during software development by contrasting agile and traditional lifecycle process models using event sequencing method for detecting activity variations among recorded performative processes in the selected projects. Our analysis shows that performative enactment of waterfall and agile ostensive routines do differ in terms of activity types carried out in the early requirements steps. However, performative routines did show conformance to ostensive specifications in iterations, affordance types, and design objects used.

Keywords: Software design, agile, sequence analysis, SDLC, Waterfall, organizational routines.

1 Introduction

A long-established design science tradition in information systems scholarship involves the creation and implementation of software development methodologies [1]. Although there are far fewer empirical assessments of information system development (ISD) methods than there are prescriptive contributions [2], comparative assessments across different methodologies do exist; however, these comparisons tend to relate the espoused methodology to organizational outcomes, often through experimental settings - rarely do these comparisons involve the comparative study of actual development practices.

* This research was supported by the National Science Foundation Grants: *VOSS-0943157* and *0943010.*

H. Jain, A.P. Sinha, and P. Vitharana (Eds.): DESRIST 2011, LNCS 6629, pp. 207–222, 2011.

One dimension along which methods have been empirically compared involves contrasting traditional methodologies to more iterative methodologies such as "rapid application development" [3] and agile methods [4]. Although much of the research comparing the two broad methodologies is anecdotal [5], there is empirical support indicating that the use of more iterative methods can result in improved outcomes such as a faster development process, better user satisfaction, and higher system quality (although results are mixed, see [6]).

The distinction between prescriptive and enacted practices has recently been fleshed out in the work of Feldman and Pentland [9], who distinguish between the ostensive and performative aspects of routines. The ostensive aspect captures the way individuals or organizations view the process, how they think about the process, and how they account for activities. The performative aspect of processes involves the situated 'carrying out' of tasks. "The ostensive aspect of the routine is the idea; the performative aspect, the enactment" [9]. In the case of software development, the ISD methodology can be considered the ostensive aspect of the software process, whereas the situated design practice is the performative aspect. In line with this, researchers have known for decades that espoused prescriptive methodologies are rarely enacted faithfully in performative design practices [7]. In line with Feldman and Pentland's view [9] of routines, the formal, planned, ostensive view of a software process is never fully reflected in the informal, situated, performative practices that enact that process model [8][9][10]. Due to this gap, prescriptive methodologies have also been criticized for their failure to guide or influence actual development processes [11]. Indeed, if methodologies are adopted at all, there is a significant variation in the way methodologies are enacted across contexts (i.e., "situational method adaptation, "[12][13]). Therefore, even though researchers have, to an extent, looked at the outcomes associated with alternative methodologies, they have rarely looked "under the hood" of performative practices to see what drives these varying outcomes. This is an important concern with respect to agile versus traditional methodologies, because on one level, all development is iterative and differences between development practices are due less to the presence of iteration and more to the way that methodologies drive distinct forms of iteration [6][14].

In this research, we look to gain a better understanding of the way traditional and agile methods are enacted within system design. In this effort, we leverage a novel socio-technical sequence notation and analysis technique [15] to compare two similar (in size and complexity) development projects in the same organization that used different methodologies: the traditional 'waterfall' and a generic agile process. The technique we adopted is rooted in the work of Andrew Abbott [16], who proposed that social inquiry needs to move beyond the identification of unidirectional relationships between generalized, static factors and instead look to reveal contextualized dynamic processes and their outcomes. Central to this process-centered view is the conception that infinite varieties of software design activities and their collections can be generated from a finite number of generative elements that make up software design activity – much like DNA produces an indefinite number of biological forms [17][18]. By studying similar projects in the same organization, we seek to address the following questions: 1) Are the performative routines observed through the enactment of the methodological artifacts indeed different? 2) If they are different, what is the shape of those differences? 3) What parts of the activities and their sequences remain the same? In addressing these questions we look to articulate and contrast how agile and waterfall methods are enacted in our study context.

The remainder of the paper is organized as follows. In the next section we review literature on software development activities and briefly introduce the traditional Software Development Life Cycle method and agile methods as ostensive routines. We then present how event sequencing can be used to analyze patterns of software development activities. Lastly, we present the findings of the event sequencing analysis and contrast the agile and waterfall methods we studied.

2 ISD Methodologies

An information system development (ISD) methodology is a set of “methods, tools, techniques and models” intended to guide and assist the software design and development process [1]. Since the mid 1960s, ISD methodologies have been introduced as ostensive routines to shape organizational responses to a given set of design tasks to avoid or mitigate the likelihood of failure in terms of quality, cost, or time [19]. An early instance of the methodology that became a widely adopted standard was first articulated by Royce, who, in the early 70’s, proposed a life cycle model to mitigate the risk of failing in complex software projects. This was based on the widely held views of good system engineering principles of the time [20] [21] [19].

This model has later become known as the ‘traditional’ or ‘waterfall’ model and can be thought of as a foundational standard for software development [19]. Waterfall is typically described as a unidirectional, top down, and non-iterative activity sequence for effectively designing software systems. The waterfall model has received criticism for a variety of reasons, including its treatment of iteration in performative routines [20] [19], and its inability to deliver cost-effective, user-driven solutions [22].

Later software methodologies have recognized iterations [23] and have given birth to less ‘monolithic’ views of activity sequences involving iterations that incrementally create designs comprise the design space [24]. Design iteration is a complex activity. It invites software designers to move back and forth between cognitive, material and representational spaces. Consequently, software designers iterate by constantly refining families of artifacts including conceptual, representational, process instantiations and methodologies [14]. The ostensive dimension of iteration is reflected in the conceptual artifacts that comprise design space and the ‘deltas’ that are incremented into design representation. The performative aspect of iteration is echoed in complex activity sequences that are repeated in various sequences during software design [14]. The concept of iteration has consequently evolved gradually from the simple idea of using a prototype as a way to learn from experience [25] to the concept that recognizes the inevitability and multiplicative nature of iterative activity [26]. In line with this idea, the repetitive use of prototypes has been extended to comprise repetition in all parts of the design activity in the form of agile or light-weight methods [27] [28]. In this regard, agile development adheres to the concept of software development as a continuous and repetitive social and technical engagement and the need to establish daily routines that gradually generate pieces of functional software. These daily and weekly routines rely on multiple technical and social techniques such as pair programming, time-boxed and test-first development to name a few [14].

Despite ample growth in the number of software methodologies, practitioners and researchers have struggled to determine the value of methodology in any given situation and what are its effects for software development and its outcomes. Analytically, no methodology is perfect, and even a light desk evaluation can easily detect pros and cons in any one of them. Any methodology is also, by definition, incomplete. Hence, one ‘size’ does not fit all and organizations need to be attentive in selecting an appropriate method for any situation [29].

In addition, due to their incompleteness and ambiguity, each methodology leaves developers significant degrees of freedom in adapting it to their design task. In this sense, the methodologies are not perfectly reflected in ongoing daily design activity [11] [7] and looking at only the methodological prescriptions would afford researchers a limited view of the enacted activity and its outcomes. This has also created significant gaps in how methods are enacted and what is the ‘acceptable’ variation of performative routines as a methodology is being followed in order to say that the methodology is still indeed being followed [12].

3 Event Sequencing

Recently, researchers in several fields have devised a variety of methods referred to as “event sequencing” to analyze ordered sequences of activities [30] [31] [32]. Originally, this method was developed by biologists to study the structure of DNA among different biological species in an effort to detect evolutionary patterns and variation within species [33]. The sequence analysis technique was adopted by social scientists in the early 90’s to study the organization of human activities such as musicians’ careers [16] [34], or spatio-temporal social behaviors [30]. These analyses, though illuminating, neither attend to generative and non-linear design tasks like software design, nor do they account for the presence of material artifacts in organizational activity; though, such artifacts are inevitably embedded into software design practices and deeply affect them [35] [36]. Gaskin and associates [15] have recently proposed an extension to this method to study variation in design activities and their elements (the ‘DNA’ of design practices), which also incorporates material artifacts (thus “socio-technical”).

We adopt this socio-technical sequencing method [15] to analyze the structure and properties of performative software development routines to reveal similarities and differences between these routines. This method is based on a process notation that offers five elements to encode variation in the elements of each design activity as to generate a systematic and rigorous representation of any performative design routine as a string of activities (see the Appendix for a detailed explanation of each element of the notation and its possible values). The elements of each activity are: (1) an *actor* containing a value for specific roles and configurations of actors; (2) an *activity* containing value for specific location and activity type; (3) an *affordance* that characterize each activity in terms of what an actor does with the tools; (4) a *tool* which offers values of the nature of materiality and type of tool; and (5) a *design object* containing value for the role of the design object in the design process and also shows its relationships with the tools [15]. The graphical notation has been implemented in MetaEdit+ software [37], which enables us to graphically encode and validate complex design routines as they occur in design projects. After collecting detailed process data through the interviews and archival data, we can thus represent the process of any software process in a visual process model (Fig. 1(a)). Each of the elements of the process model is then assigned a code according to their value in the design 'taxonomy' (Appendix 1), thus converting the graphical sequence into a concatenated string of alphanumeric characters where each string represents one activity or activity sequence. We use the MetaEdit+ query language and Excel scripts to extract and generate such event sequence presentations from the visual process descriptions. Finally, we analyze these strings to derive descriptive statistics. We use the ClustalG sequencing software to determine proximity and distances between processes [31] [32] [38] on activity, or any activity sequence levels (see Fig.1 (b)).

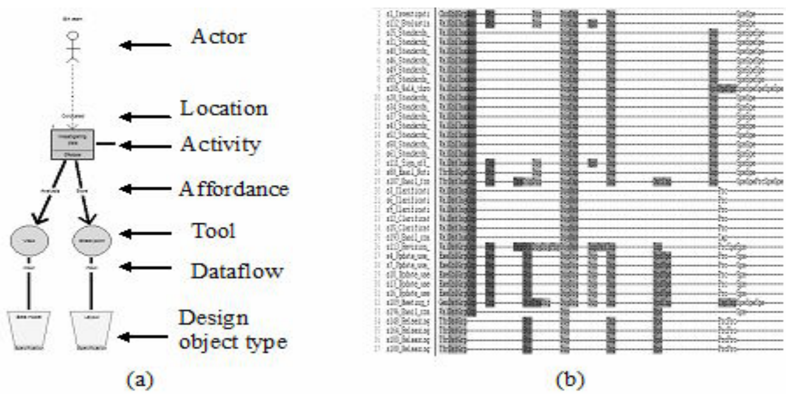


Fig. 1. (a) Snapshot view of activities in metamodel (b) Sequence alignment in one project

4 Case Study of Waterfall and Agile in a Large Automotive OEM

We collected process data within the software development unit of a large multinational automobile manufacturing organization, Beta. This unit focuses on developing and integrating the software that organizes complex product information and the associated processes for design and manufacturing. We selected one traditional waterfall project and one agile project of comparable scale and complexity that were developed during roughly the same time period using roughly the same sized team, and overlapping with many of the same artifacts and other infrastructural elements. Both projects had a size of c.a. 20 man-years and lasted for c.a. 2 calendar years. They were both important, strategic initiatives receiving significant management attention and, consequently, were designed and implemented with highly competent and experienced talent. Although not a perfectly controlled experiment, the projects were selected with the idea that the main difference between the projects would be the explicit focus on using an agile methodology for one project and a waterfall method for another project, thus offering a possibility to conduct a sort of quasi-experiment or natural experiment [39].

BOM Search Project (Waterfall): The Bill of Material (BOM) search project followed a traditional waterfall structure as dictated by Beta's life cycle development methodology that is founded on object oriented data modeling, use cases, and derivation of a design architecture based on object oriented design. The project was initiated in the first quarter of 2009 to enhance search in the Bill-Of-Material (BOM) database and it lasted for about two years. It is relatively large in size (over 20 man years) and involved 24 people working in two locations (U.S. and India). The project followed the following four phases as dictated by the waterfall methodology:

(1) ***Requirements*:** In this phase, use cases were formulated. The creation of use cases started off by investigating data (data in the data model and business logic) with tools like Visio and SharePoint by the team. The team frequently interacted with the Subject Matter Expert (SME) (the businessperson or the user who will be using the BOM search) to get a clarification on the use cases prepared (there were three use cases prepared). At the end of the stage, the use cases were frozen and were passed on to the next stages as specifications of system behaviors and related requirements. The team used tools like RE Pro (to record the use cases), Outlook and the telephone to communicate the requirements.

(2) ***Data model creation and testing*:** This phase partially overlapped with the use case generation. The initial data model was iteratively revised by taking into consideration the use cases and data modeling standards. The data model was developed using multiple tools like Toplink, WS Corps (which is a framework that supports the firm's website), persistence code (which is a framework which can communicate with the Toplink). The data model underwent a series of revisions while defects were generated in relation to use cases.

(3) ***Validation, Design & Development*:** After the development of the implementation mode,l where the generic data model is transformed based on the business logic and use cases to create an implementation model are created, the team in the U.S. met to inspect all of the use cases and the data model. The model was tested for defects, consistency, and whether it met the internal Java standards. The design and implementation cost were then estimated and discussed with the offshore team. After finalizing the model, the code implementation started in the offshore site, which used web-based tools to coordinate the evolving design with the local team.

(4) ***QC testing and version release*:** Once the code was written, it was internally tested against use cases. The code was validated using the use cases and defects were reported to the offshore team. After the code was fixed, the BOM search interface was released into the production.

LCM Project (Agile): This project addressed how the BOM database deals with engineering specification changes. The project has now been running for a few years and the software team creates a new release every three months with patches in between. We specifically investigated the design of the 1.5 and 1.6 releases referred to as "light change management" or LCM. The 1.5 release began in September of 2009 and went live with the release of 1.6 in January 2010. The project involved nearly 20 designers and a large number of lead users in OEM locations in the U.S. and Europe. The development team chose to use a generic agile process for developing this application that was not strictly based on any particular method. Rather, they used aspects of SCRUM and eXtreme Programming. The software progress and deadlines are reassessed daily and changes are made as necessary. Thus, everyone involved is always knowledgeable about the status of the application and the deadlines. The following is a short illustration of the main elements of the technique:

(1) ***Planning***: An internal team writes the requirements and posts them on an internal Wiki. These requirements can be anything from simple cosmetic tweaks to the existing software to completely new functionality. Once the requirements are received, release planning starts. The release cycle is around three months. The project engineers know that all releases are smaller parts of a larger project; therefore, all releases are just small iterations of a larger project but typically include several man-years of software development.
(2) ***Managing***: The LCM developers work in an open space where they are in constant contact with one another. Each day starts with a meeting at nine in the morning, followed by one at ten in which progress is checked, tasks are assigned and re-examined, and developers are moved around based on the project status and task needs. There is no ownership of specific tasks, rather, every day someone new may be working on a different task.
(3) ***Designing***: From a design standpoint, LCM has created its own systems metaphor, which it adheres to through all projects. Additionally, they only implement the functionality, which is necessary for the current release. The code is not refactored throughout the release process.
(4) ***Coding***: Beta's LCM team is very close to its customers, both in physical proximity and in regards to the project. A member of their customers is always with them throughout the day. Unit tests are developed before any code is written. They have strict procedures for testing, maintaining, and deploying the code. Additionally, one person is in charge of deploying, one person is in charge of QC, and one person is in charge of code review.
(5) ***Testing***: As for testing, all code has unit tests, which it must pass before it can be deployed. These test cases are written before any of the code is actually developed. Bugs are found before the code is deployed and bug fixes are sent back to the developer that wrote the code.

To collect process data about these projects we conducted in-depth interviews with project managers and team members. We visited the company site where the IT department is located on four occasions. We also interviewed process managers in the Indian site by phone on two occasions. The transcribed interviews were then converted into a graphical process model, which then underwent thorough process reviews and subsequent validation with the software team leaders. We engaged in 22 interviews of at least an hour to collect and validate the data.

5 Research Findings

Figure 2 depicts full process models for both projects to give the reader an understanding of the scale and complexity of projects and the resulting process models. We analyzed these process models for variation in activities, the nature and scope of iterations, and the difference in distributions across activity types, affordances and the distribution of different instances of activities to find out what are the true performative differences between the two types of software processes.

We first carried out a sequence analysis and alignment step to detect the spread of activity variation in both projects. This resulted in the overall clustering of activities for both process models based on their similarities. These clusters are visually depicted and interpreted in Figures 3, 4 and 5 based on the activities each cluster contains.

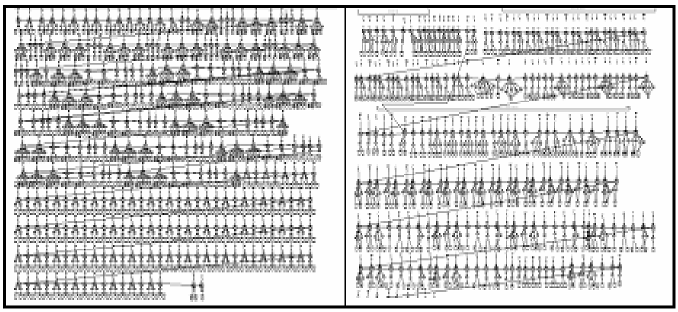


Fig. 2. Sequence flow of activities in agile (L) and waterfall(R)

Through a visual assessment of each cluster model, it is clear that the agile method is simpler in terms of the number of types of activities involved. At the project level, agile has three validation clusters and two limited negotiation clusters in addition to the several execution clusters. The waterfall process, on the other hand, has far more activity variety, with a great deal more "miscellaneous" clusters that did not fall neatly into the key activities of execute, validate, and negotiate. This is revealing as it shows that the waterfall method as a bit 'messier' in terms of activity variance as it includes a larger number of control, coordination and other types of support activities. In the agile method there also appears to be a cleaner, more straightforward delineation of the main activities. We also analyzed the variation in the activities in the front end and the back end of the process models to detect to what extent the early and late phases of the two models differ.

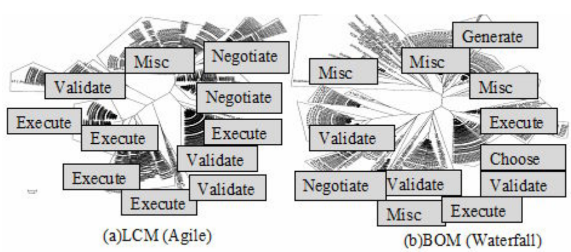


Fig. 3. Project level clustering of the activities

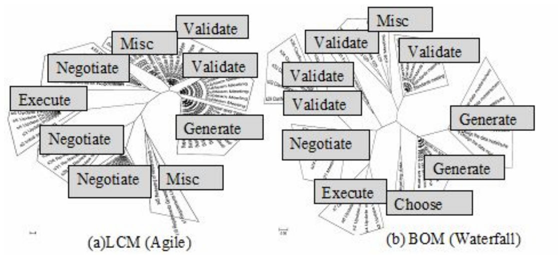


Fig. 4. First phase clustering of the activities

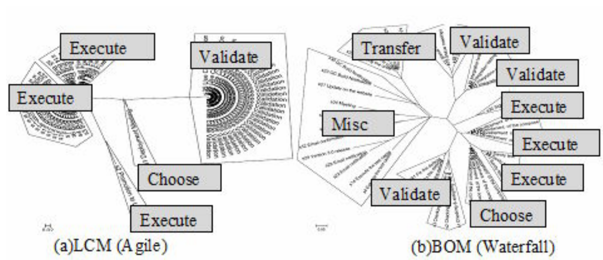


Fig. 5. Final phase clustering of the activities

To this end, we analyzed the first 20% and the last 20% of the activities as we expected these parts to represent the most likely variation while capturing large enough portion of the whole process. Again, the phase analysis showed higher number of activities and related branches for the waterfall project. In the early stages, the agile project also greatly relied on negotiation activity, while the waterfall project relied on validation. The analysis of the implementation and roll out phase revealed a greater dependency on validation and execution activities within the agile method. The agile method also relied heavily on validation in the final phase.

To analyze differences in iteration in the two processes, we tallied the number of iterated activities and the number of iterated design objects in both processes. As expected, the agile process had more iterations proportionally in the overall sequence and its iterations were smaller (Table 1). The granularity of iteration was smaller with the agile process (Fig 6). Not surprisingly, the waterfall process also evidenced a good deal of iteration.

Table 1. Iterations in the LCM(agile) and BOM (waterfall) model

	Agile	Waterfall	Absolute variation (%)
% of iterated activities	97.37%	85.5%	11.87
% of iterated design objects	98.4%	78.09%	20.31

As Figure 7 indicates, the waterfall process had many activities that were not iterated at all, whereas the agile process did not have many activities, which were not part of some iteration cycle. In short, more detailed iterations were occurring more frequently (see also Fig 8 and Fig 9). Note also that the agile method's activity frequencies are more skewed approaching a power distribution. The line on the chart indicates the proposed distribution equation that best fits the data suggesting (Figures 8 and 9) that a Beta distribution offers the best fit for the observed frequencies.

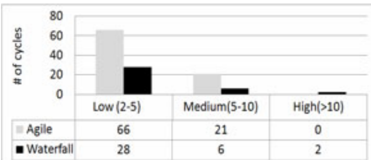


Fig. 6. Frequency of types of iteration

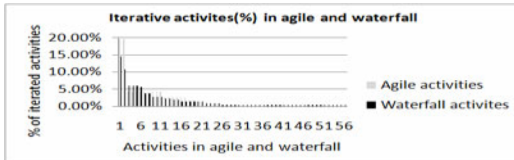


Fig. 7. Frequencies of iteration

Next we asked the extent to which activity types that underlie our model (choose, execute, negotiate, transfer, validate) were proportionally present in the activity sequence. We found that both the agile and waterfall processes had some instances of each type of activity, except that agile had no activities explicitly oriented towards choice (choice is embedded in generation). Further, waterfall had more exclusive negotiation-oriented activities, whereas agile had relatively more activities focused on execution and validation (Fig. 10).

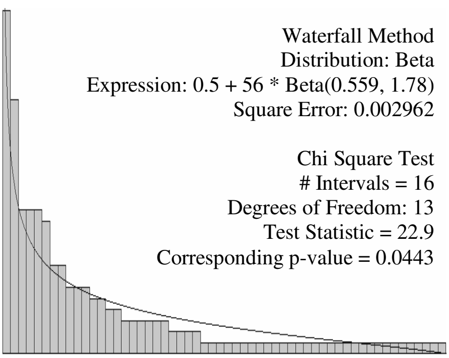


Fig. 8. Activity distributions BOM (waterfall)

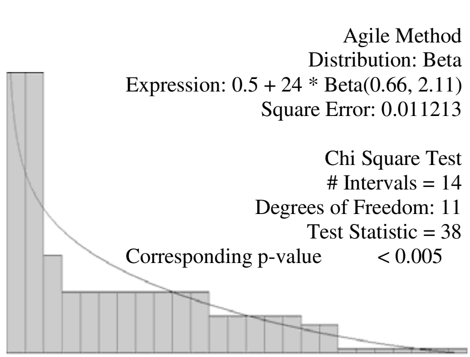
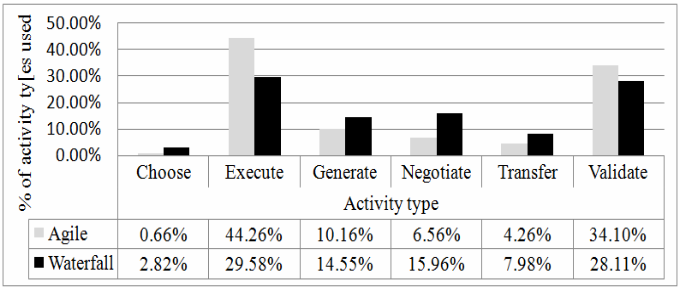


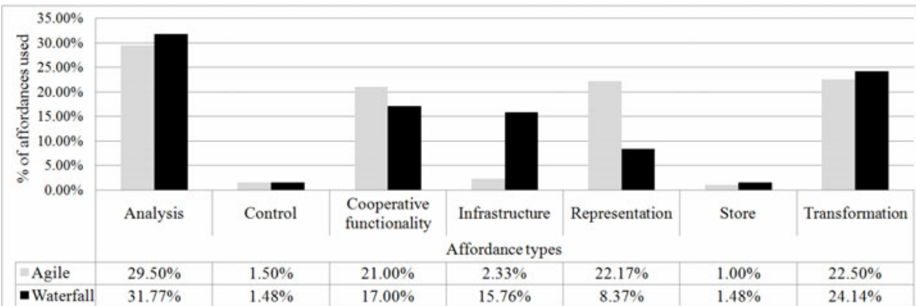
Fig. 9. Activity distributions LCM (agile)

We also looked at the distribution of affordances, i.e., how various tools were enacted in the process. In general the two processes looked quite similar (Fig. 11). Exceptions were that waterfall made more use of infrastructural elements of the tools and the agile process made use of significantly more representational affordances relative to the total project.



	Choose	Execute	Generate	Negotiate	Transfer	Validate
Agile	0.66%	44.26%	10.16%	6.56%	4.26%	34.10%
Waterfall	2.82%	29.58%	14.55%	15.96%	7.98%	28.11%

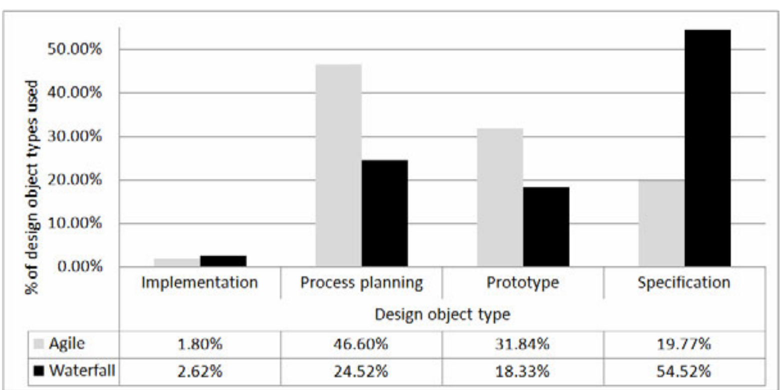
Fig. 10. Activity types in LCM (agile) and BOM (waterfall)



	Analysis	Control	Cooperative functionality	Infrastructure	Representation	Store	Transformation
Agile	29.50%	1.50%	21.00%	2.33%	22.17%	1.00%	22.50%
Waterfall	31.77%	1.48%	17.00%	15.76%	8.37%	1.48%	24.14%

Fig. 11. Affordance types in LCM (agile) and BOM (waterfall)

The final part of the analysis involved analysis of the variance in the design objects (Fig. 12). The waterfall method involved more uses of specifications, whereas the agile method involved more prototyping (i.e. developing partial implementations). However, the waterfall method did also have a good deal of prototyping, and the agile method did involve some uses of specifications. However, one main difference is that the agile process involved a markedly greater amount of process planning due to the frequent and ongoing nature of the daily and weekly planning.



	Implementation	Process planning	Prototype	Specification
Agile	1.80%	46.60%	31.84%	19.77%
Waterfall	2.62%	24.52%	18.33%	54.52%

Fig. 12. Design object types in LCM (agile) and BOM (waterfall)

6 Discussion and Conclusion

In this section we review our original question: whether the enacted methodologies really differ, and if so, how? Our analysis shows that the enactments of the agile and waterfall methodologies are substantially different. Thus, we might conclude that ostensive methods matter and there are indeed significant differences in how activities are organized and what activities are carried out in the waterfall and the agile process, respectively. This is shown in the iterations, types of activities proportionally enacted, the affordances carried out, and the frequency distribution of activities. At the same time, both the waterfall and agile process show dependency on a similar set of activity types though their spread is much wider with the waterfall method. There are also many more "mixed" activities in the waterfall process. We also show that the early phases of the agile method rely heavily on negotiation and validation whereas the waterfall method does not. Likewise, in the later phases, validation is a dominant activity for agile, whereas this is not the case for waterfall. Due to space limitations we cannot elaborate extensively on our findings, but some highlights follow:

Our first finding suggests that the waterfall method is "messier" than it is typically characterized. The waterfall method has a greater number of complex activities, that is, there are more tools and objects used in each activity than are commonly credited. Our findings also confirm that agile methods involve relatively more iterations. Agile methods are *extremely* iterative in terms of how the activities are organized in the sense that agile has few activities that do not in some sense iterate. However, what is surprising is that it was not as if the agile method was iterative and waterfall was not. Both were *highly* iterative, indicating the need to differentiate between forms of iteration. Another finding was that most tool affordances were identical between the two methodologies. Thus, on some level, *a software process is a software process* and this implies a certain way of enacting technologies and tools. What was interesting was the frequent use of representations to engage clients in the agile method. Agile is not typically characterized as a representationally intensive process; rather, it is thought of as more of a coding marathon. Our results indicate otherwise, that agile is a highly representation driven process. This shows that companies can appropriate agile methods to fit their culture and way of doing things with their clients. Next, the waterfall process involved more of the use of specifications while agile involved more of the use of prototypes. What may come as a surprise is that agile did indeed involve a good deal of specification, while waterfall also had a good deal of prototyping. Finally, our findings indicate that the enactment of software methodologies differ from the idealized ostensive routines implied by their methodological descriptions. It should be noted; however, that enactments of these routines are influenced by a large number of exogenous factors like culture, business environment and people, which were not accounted for in this short analysis.

References

1. Hirschheim, R., Klein, H., Lyytinen, K.: Information systems development and data modeling: conceptual and philosophical foundations. Cambridge Univ. Pr., Cambridge (1995)

2. Wynekoop, J., Russo, N.: Studying system development methodologies: an examination of research methods. Information Systems Journal 7(1), 47–65 (1997)
3. Gordon, V., Bieman, J.: Reported effects of rapid prototyping on industrial software quality. Software Quality Journal 2(2), 93–108 (1993)
4. Cockburn, A., Highsmith, J.: Agile software development, the people factor. Computer 34(11), 131–133 (2002)
5. Lindvall, M., Basili, V., Boehm, B., Costa, P., Dangle, K., Shull, F., et al.: Empirical findings in agile methods. In: Wells, D., Williams, L. (eds.) XP 2002. LNCS, vol. 2418, pp. 197–207. Springer, Heidelberg (2002)
6. Berente, N., Lyytinen, K.: Iteration in systems analysis and design: Cognitive processes and representational artifacts. Sprouts: Working Papers on Information Environments, Systems and Organizations 5(4), 178–197 (2005)
7. Parnas, D., Clements, P.: A rational design process: How and why to fake it. In: Formal Methods and Software Development, pp. 80–100 (1985)
8. Brown, J., Duguid, P.: Organizational learning and communities-of-practice: Toward a unified view of working, learning, and innovation. Organization Science 2(1), 40–57 (1991)
9. Feldman, M., Pentland, B.: Reconceptualizing Organizational Routines as a Source of Flexibility and Change. Administrative Science Quarterly 48(1), 94–121 (2003)
10. Suchman, L.: Plans and situated actions: The problem of human-machine communication. Cambridge UP, Cambridge (1987)
11. Curtis, B., Krasner, H., Iscoe, N.: A field study of the software design process for large systems. Communications of the ACM 31(11), 1268–1287 (1988)
12. Russo, N., Wynekoop, J.: The use and adaptation of system development methodologies. In: Managing Information & Communications in a Changing Global Environment. Idea Group Publishing, PA (1995)
13. Smolander, K., Tahvanainen, V., Lyytinen, K.: How to combine tools and methods in practice—a field study (1990)
14. Berente, N., Lyytinen, K.: What is being iterated? Reflections on iteration in information system engineering processes. In: Conceptual Modelling in Information Systems Engineering, pp. 261–278 (2007)
15. Gaskin, J., Lyytinen, K., Thummadi, V., Schutz, D., Yoo, Y., Weiss, A., et al.: Sequencing Design DNA: A Set of Methodological Artifacts for Sequencing Socio-Technical Design Routines. In: International Conference on Information Systems, St. Louis (2010)
16. Abbott, A.: Sequence Analysis: New Methods for Old Ideas. Annual Review of Sociology 21(1), 93 (1995)
17. Abbott, A.: A Primer on Sequence Methods. Organization Science 1(4), 375–392 (1990)
18. Pentland, B.T.: Sequential Variety in Work Processes. Organization Science 14(5), 528–540 (2003)
19. Sommerville, I.: Software process models. ACM Computing Surveys (CSUR) 28(1), 269–271 (1996)
20. Fitzgerald, B.: Systems development methodologies: the problem of tenses. Information Technology & People 13(3), 174–185 (2000)
21. Royce, W.: Managing the development of large software systems (1970)
22. Lyytinen, K.: A taxonomic perspective of information systems development: theoretical constructs and recommendations (1987)
23. Boehm, B., Turner, R.: Balancing agility and discipline: A guide for the perplexed. Addison-Wesley Professional, Reading (2003)
24. Graham, D.: Incremental development and delivery for large software systems (2002)

25. Mathiassen, L., Seewaldt, T., Stage, J.: Prototyping and specifying: principles and practices of a mixed approach. Scandinavian Journal of Information Systems 7(1), 55–72 (1995)
26. Boehm, B.: A spiral model of software development and enhancement. Computer 21(5), 61–72 (1988)
27. Paetsch, F., Eberlein, A., Maurer, F.: Requirements engineering and agile software development (2003)
28. Tomayko, J.: Engineering of unstable requirements using agile methods (2002)
29. Laplante, P., Neill, C.: Opinion: The demise of the waterfall model is imminent. Queue 1(10), 10–15 (2004)
30. Shoval, N., Isaacson, M.: Sequence Alignment as a Method for Human Activity Analysis in Space and Time. Annals of the Association of American Geographers 97(2), 282–297 (2007)
31. Wilson, C.: Activity Patterns of Canadian Women: Application of ClustalG Sequence Alignment Software. Transportation Research Record: Journal of the Transportation Research Board 1777(-1), 55–67 (2001)
32. Wilson, C.: Reliability of Sequence-alignment Analysis of Social Processes: Monte Carlo Tests of ClustalG Software. Environment and Planning A 38(1), 187 (2006)
33. Durbin, R., Eddy, S., Krogh, A., Mitchison, G.: Biological sequence analysis. Cambridge university Press, Cambridge (2002)
34. Abbott, A., Hrycak, A.: Measuring resemblance in sequence data: An optimal matching analysis of musicians' careers. American Journal of Sociology 96(1), 144–185 (1990)
35. Leonardi, P.M., Barley, S.R.: Materiality and Change: Challenges to Building Better Theory about Technology and Organizing. Information and Organization 18(3), 159–176 (2008)
36. Orlikowski, W.J., Scott, S.V.: Chapter 10: Sociomateriality: Challenging the Separation of Technology, Work and Organization. The Academy of Management Annals 2, 433–474 (2008)
37. Tolvanen, J., Rossi, M.: MetaEdit+: defining and using domain-specific modeling languages and code generators (2003)
38. Wilson, C., Harvey, A., Thompson, J.: Clustalg: Software for analysis of activities and sequential events. Paper Presented at the Workshop on Sequence Alignment Methods, Halifax (October 2005)
39. Shadish, W., Cook, T., Campbell, D.: Experimental and quasi-experimental designs for generalized causal inference (2002)
40. Markus, M.L., Silver, M.S.: A Foundation for the Study of IT Effects: A New Look at DeSanctis and Poole's Concepts of Structural Features and Spirit. Journal of the Association for Information Systems 9(10), 5 (2008)

Appendix: Taxonomy of the Design Components

Design component	Items	Description
Activity Type Activity type refers to the purpose of the design activity.	Generate	Action oriented planning and creativity-driven tasks such as brainstorming, coming up with plans, or producing something as a design.
	Transfer	Transferring information or objects between people or locations.
	Choose	Picking a correct or preferred option or answer. Coming to consensus.
	Negotiate	Resolving policy and payoff conflicts.
	Execute	Performing or executing a plan- producing an object according to a plan or a design
	Validate	Verifying quality and consistency.
Actor Configuration Actor configuration refers to the number and grouping of the actors involved in the activity.	1 individual	Single individual
	1 group	A group of individuals with a single functional purpose
	many individuals	More than one individual, each with a separate functional purpose
	many groups	More than one group, each with a separate functional purpose
	individuals and groups	A mix of both individuals and groups, each with a separate functional purpose
Tool Materiality Tool materiality simply refers to the material makeup of the tool being used for a particular design task.	Physical	The material nature of the functional aspects of the tool is physical, rather than digital. For example, the functional aspect of paper (ability to represent information) is physical.
	Digital	The material nature of the functional aspects of the tool is digital, rather than physical. For example, a word processing document (ability to represent information) is digital.
Tool Affordance Affordances refer to “the possibilities for goal oriented action afforded by technical objects to a specified user group understood as relations between technical objects and users and understood as potentially necessary (but not necessary and sufficient) conditions for "appropriation moves" (IT uses) and the consequences of IT use” ([40] p. 622).	Representation	Functionality to enable the user to define, describe or change a definition or description of an object, relationship or process
	Analysis	Functionality that enables the user to explore, simulate, or evaluate alternate representations or models of objects, relationships or processes
	Transformation	Functionality that executes a significant planning or design task, thereby replacing or substituting for a human designer/planner
	Control	Functionality that enables the user to plan for and enforce rules, policies or priorities that will govern or restrict the activities of team members during the planning or design process
	Cooperative	Functionality that enables the user to exchange information with another individual(s) for the purpose of influencing (affecting) the concept, process or product of the planning/design team
	Support	Functionality and associated policy or procedures that determine the environment in which production and coordination technology will be applied to the planning and design process
	Infrastructure	Functionality standards that enable portability of skills, knowledge, procedures, or methods across planning or design processes
	Store	Functionality that allows information to be housed within a device.

Activity Location Location refers to where the design activity takes place.	Collocated	Actors are located in close proximity to each other at headquarters during the design activity.
	Distributed	Actors are distributed during the design process.
	Remote Collocated	Actors, though located in close proximity to each other, are not at headquarters during the design activity.
	Remote Distributed	Actors are distributed and not at headquarters during the design activity.
Design Object Type Design object type refers to the purpose of the design object being used as an input, being updated, or resulting as an output of a design activity.	Specification	The design object is instructions for design product parameters and constraints.
	Design	The design object is a physical or digital prototype of part or the entirety of the intended eventual design product. This design object is used for further analysis and representation.
	Implementation	The design object is actually used to complete, in part or whole, the intended eventual design product.
	Process planning	The design object is instructions for future design activities.
Tool-Design Object Connection	Output	The data flow when the design object did not exist prior to the task, but was created during the task
	Input	The data flow existed prior to the task, but did not change during the task
	Update	The data flow existed prior to the task AND did change

e-wallet Prototypes

Mia Olsen, Jonas Hedman, and Ravi Vatrapu

Department of IT Management, Howitzvej 60, 2000 Frederiksberg, Denmark
miao_86@yahoo.dk, {jh.caict,rv.caict}@cbs.dk

Abstract. The design outcomes of this paper are four e-wallet prototypes. E-wallets are intended to replace the existing physical wallet, with its notes, coins, bills, photos, plastic cards, loyalty cards etc. Four different user groups, including Young Teenagers, Young Adults, Mothers and Business Men, have been involved in design and test of the prototypes. Interviews and user tests have provided data for the construction of first a conceptual model, in the form of sketches, and later a functional model, in the form of mock-ups. During the design phases, knowledge was gained on what properties, including design, functional, service, and interaction, the user groups would like the e-wallet to hold. The properties have been to develop four prototypes, one for each user group.

Keywords: e-wallet, cashless society, design science, prototype.

1 Introduction

The digital revolution continues to transform most aspects of our daily life. In particular, the digital revolution has resulted in the vertical convergence of business channel capacities [12]. The digital revolution also continues to transform the public sector organizations and services. For instance, the Danish public citizen portal called borger.dk which forms an online entrance to the public sector with access to public information and digital self-services concerning topics such as family and children, taxation, residence and buildings, and disabilities. Other examples are digital bus tickets bought via mobile phones, online purchases, and social interactions made via SMS, emails and social networks. A next step in the digital revolution is the transformation of the time honored traditional physical wallet into the e-wallet.

The focus of this paper is on mobile payments. And when everything else is going mobile, the payments have to be mobile too irrespectively of time and place. So, there is a need for an electronic wallet – an e-wallet. There are many mobile and wireless payment solutions, but most of them have failed or their adoption rate has been lower than expected. Dahlberg et al. [4] suggest that technological development of such solutions should be directed towards a closer cooperation with users, and Mallat [13] stresses that future mobile payment research should focus on usability, as this is an unexplored area of mobile payments. Set within this context, the purpose of this paper is to propose e-wallet prototypes.

2 Background

In the beginning of the 2000's, early mobile content and services, such as ring tones and logos found market success and made mobile payment services a hot topic. At that time,

H. Jain, A.P. Sinha, and P. Vitharana (Eds.): DESRIST 2011, LNCS 6629, pp. 223–236, 2011.

mobile payments were commonly perceived as the killer application for mobile commerce [17, 25]. Later, mobile payments were suggested as an alternative for micro-payments at point-of-sales, where the use of cash had been declining for many years [9]. Even though cash is still growing due to the economic growth. Lots of mobile and electronic payment solutions have been introduced ever since, but as already mentioned most of them have failed or have had a low penetration rate [3, 13]. Moreover, payment is an institutionalized basic act, which cannot be changed just like that [10]. Payment is transacted in almost the same way worldwide, and it would cause problems if each country had its own electronic payment system [5]. Further issues arise when companies additionally develop their own electronic payment systems, such as those for public transportation and retail chains. There is a need for standardization of mobile payments [14].

One of the more successful new standardized electronic payment systems is PayPal. Initially PayPal enabled people to perform transactions of small payments by means of e-mails. Since then, PayPal's system has been re-designed and extended several times Today PayPal has more than 220 million accounts and is experimenting in the area of mobile payments through a partnership with a start-up that provides stickers for mobile phones that can link the phone to some special payment terminals in the stores.

Two other successful electronic payment systems are the Oyster Card in London [11], and the Octopus Card in Hong Kong [17]. The question is whether their success is due to the fact that they initially were introduced to support mass transit systems, instead of trying to substitute all payments. The Octopus Card has later been extended to include payments at convenience stores, fast-food restaurants, supermarkets, parking meters, car parks, vending machines and service stations.

In Africa a new kind of mobile payment was introduced in the beginning of 2007. The payment system is called M-PESA ('M' is for mobile and 'PESA' is the Swahili word for cash) and was developed by Kenya's largest mobile network operator Safaricom, which is a part of the Vodafone Group. Within the first week more than 20,000 M-PESA accounts had been registered and two years later in 2009 the number of accounts had reached six million.

Visa and MasterCard are introducing new payment systems, including Visa PayWave and MasterCard PayPass. Both solutions are contactless payment technologies, which uses RFID-technology, so that the payer does not have to swipe the card or insert it into another device to pay. They were primarily introduced as smart card technology, but have since been extended to include key fobs and Near Field Communication (NFC) enabled mobile phones.

Similar solutions exist already in Japan where many daily transactions are made with mobile phones. In 2004 Sony, NTT DoCoMo, and local banks formed a joint venture and launched a mobile payment system. The system is based on Sony's chip FeliCa and allows customers to use their mobile phones as credit cards, access cards, fare tokens on all kinds of public mass transit, and several other utilities.

So, while the most popular payment instruments are cash, cheques, and debit and credit cards [4] with smart cards being the most serious challenger to traditional cash [2], the ways to make contactless payments and especially mobile payments are increasing. Dahlberg & Mallat [2] furthermore concludes that the reasons for using electronic payment channels are advantages such as accessibility, convenience, speed, privacy and control, and that electronic payments are preferred in simple routine service transactions.

When looking into the future, companies and experts agree that the mobile phone is the technical device that they will try to turn into the new wallet, mainly because of the number of mobile phones in use, which no other technical device can match, but also due to the fact that most of us carries our mobile phones with us all the time. If the mobile phone becomes a common platform for digital payments it is likely that some traditional payment instruments will decrease and even vanish. But it is also a possibility that the mobile wallet will just become a new way of entering the current card and account-based payment services [4].

Overall, it can be said about the future, that mobile payments may not be able to substitute the traditional physical wallet and may therefore just complement existing payment solutions. But, it is also a possibility that a substitution will happen gradually and first after several years be completed [13]. No matter what will happen, exploring what properties the users find useful in a mobile wallet is a step on the way towards a cashless society [6].

3 Methodology

The choice of method was driven by the research problem, which is the development of e-wallet prototypes with focus on the interaction between user and artifact. The focus on human-computer interaction leads to issues that are complex and grounded in multiple disciplines. Consequently, questions frequently arise that have a thin or no theoretical background, and exploring these, is where Design Science Research – exploring by building – proves useful [22].

3.1 The Design Process

There are several guidelines and approaches on how to proceed with design science projects [7, 18, 23]. In this project we draw upon Takeda et al.'s [21] design science model. The choice is motivated by that Takeda et al. (1990) was one of the earliest to structure and formalize the process of using Design Science. The model is also applied in recent design science literature, such as Vaishnavi & Kuechler [22].

Takeda et al. [21] describes a process starting with an Awareness of Problem phase. The awareness of a problem typically comes from wonder or a problem in current practice that the researcher aims to solve. The output of this phase is a description of the problem and a proposal for researching this problem. In the next phase, the Suggestion phase, solutions are found in existing knowledge, followed by an attempt to implement an artifact based on the suggested solution (called the Development phase). Knowledge in the Suggestion phase refers to solutions from other areas, theories, or ideas from potential users. Afterwards, in the Development phase, an attempt at developing and implementing an artifact according to the suggested solution is performed. In this phase most of the design work takes place. The techniques for implementation vary, depending on the artifact. The implementation itself can be very ordinary and does not need to involve innovation beyond the state-of-practice for the given artifact; the innovation is in the design, not the construction of the artifact.

The output of this phase is findings about the artifact's application and functionality. Afterwards the Evaluation phase commence where the implementations of the solution is evaluated, and finally, a Conclusion phase concludes the design cycle indicating that the design project is finished by deciding that the results are "good enough", and by summarizing what the contributions of the artifact are. The phases Development, Evaluation, and further Suggestions are iterative until the results are "good enough" or saturation has been reached.

3.2 User Involvement and Data Collection

The users involved in this project were mainly found at Facebook among peripheral acquaintances and friends of friends, in order to keep prior knowledge of the interviewees to a minimum and minimize biases. A further selection criterion for the interviewees was the degree of use of technology in their everyday lives, as this was estimated to be necessary in order for the interviewees to be able to understand the mobile wallet concept. The number of users was 26 for the Suggestion phase and 16 for the Evaluation phase. Table 1 provides information on the number of participants and demographics.

Table 1. Participants involved

User groups	No. of part.	Age	Time Period (2010)	Location
		Suggestion phase		
Young Teenagers	8	13-15	Sep-Dec	Home and school
Young Adults	8	19-25	Sep-Dec	Home and library
Mothers	5	32-37	Sep-Oct	Home and workplace
Business Men	5	46-53	Sept-Dec	Home and workplace
		Evaluation phase		
Young Teenagers	4	15	Jan	Home
Young Adults	4	20-22	Jan	Home
The Mothers	4	30-37	Jan	Home
Business Men	4	46-53	Jan	Home and workplace

The participants for both phases represented four different user groups: Young Teenagers (YT), Young Adults (YA), Mothers (M) and Business Men (BM). The reason for choosing these four user groups is the fact that they loosely cover the phases of Wells and Gubar's [24] widely used consumer life cycle. Furthermore, one of our underlying assumptions is the need for multiple solutions from different user groups. The interaction time between researcher and user varied from 15 to 60 minutes and were conducted in the autumn of 2010. To avoid the issue of the artificial environment intimidating the interviewees, the interviews were held at a place chosen by them, mainly their residence or work place.

4 The Design Process e-Wallet Prototypes

4.1 The Awareness and Suggestion Phase

The starting point of the design project was the identified lack of e-wallets or any conceptual models. The problem was grounded both in the literature and in the practice (expert interviews). The Suggestion phase is where the work with the proposal from the previous phase is initiated. The output of this phase is a tentative design. The practical work with the suggestion took its starting point in the users. We recruited participants from four user groups based on the assumption that the groups would differ from each other, regarding their needs and expectations to the wallet.

Munck [15] states, that the understanding of end-users behaviors and needs is a success criterion for contactless and mobile payments. The Suggestion phase involved fours steps: Usability Goals and User Experience Goals, Personas, Sketching, and Scenarios.

Usability Goals and User Experience Goals. If the primary objective of developing a product for a group of users is made clear, it is easier to understand these users. Classifying the objectives in terms of usability goals and user experience goals can do this. Usability goals are concerned with meeting specific criteria of usability, whereas user experience goals are concerned with developing user experiences [20]. However, as this project only focused on what prototypes the users needed in the wallet and not on the experience they had using it, the usability was the focal point while user experience goals were not written. Yet, it is important to note, that the two kinds of goals are not clearly separable, since one of the goals is fundamental to the other. But, since this project is an exploratory study that forms the basis for future research, it is acceptable that not all perspectives of the wallet are covered. The following four overarching goals were identified during the first round of interviews:

- Efficiency: Carrying out a common task such as paying with the mobile wallet, should imply no more than six steps, which is the number of steps it takes to pay with a payment card today (take the card out of the wallet – place it in the payment terminal – type the PIN – click OK – remove the card from the terminal – put it back in the wallet). This usability goal was chosen as it was pointed out as important by Dahlberg & Mallat [2], and additionally mentioned by one of the interviewees.
- *Safety*: It should not be possible to make a payment by mistake. This goal was chosen, since security is important according to almost half of the interviewees and Dahlberg et al. [2].
- *Utility*: The mobile wallet should provide an appropriate set of functions that will enable users to carry out their conventional tasks from the physical wallet, in the way they want to do them. This was chosen as a criterion for usability because of the fact that the interviewees had so many different ways of using their wallets.
- *Learnability*: It should be possible for the user to work out how to use the mobile wallet by exploring the interface. This is important, as people do not like spending a long time learning how to use a new system, and two of the

> interviewees told that they do not read instruction manuals. Learnability is especially important for interactive products intended for everyday use [20], which includes the mobile wallet.

Personas. After having conducted the interviews and written the usability goals, four personas were created representing the four user groups. A persona is a thorough description of a typical user of the system that is developed. Hence, the designers can focus on designing the system to this user, rather than to a whole group of users. A persona is not a description of a specific person who exists in reality, but a mixture of an amount of user data [20]. Generally, these descriptions are called fictitious user descriptions, and many theorists have worked with different designations for the process of working with the fictitious users. The creation of the personas in this project followed the proposed structure in the second phase of The Persona Lifecycle, which focuses on persona conception and gestation. The Persona Lifecycle has been widely cited, among others by Nielsen [16]. Below we illustrate the persona constructed for the Business Men user group:

> Tom is 51 years old and works as IT Manager for a large international company. He is married to Susanne with whom he has two children – an 18 year-old boy named Jacob, and a 15 year-old girl named Stephanie. The family lives in a large house in a residential neighborhood. He is a very balanced person although he has got a lot on his plate, and both his colleagues and family love him for that. He never rushes about but keeps calm and always has a good grasp of the situation. He can be a hard leader and father when necessary but otherwise he is nice and pleasant. He likes having goals in life as they give him something to get up to in the morning, and he feels satisfied when he reaches them. He uses technology extensively. He has a laptop and a smart phone from work, which he uses both for work related, and private stuff. The smart phone is a BlackBerry and is used mainly for work related calls and emails, and news reading. He has got profiles on Facebook, LinkedIn and a social network only for the employees of the company. He doesn't log on to these networks very often, but when it happens it is mostly via his BlackBerry. When he's on the go Tom enjoys listening to music on his iPod.

Sketching. After having gained knowledge of the cashless society and the future users of a mobile wallet, the design of the mobile wallet was conceptualized. This was done through a sketching process that started off when the interviewees from the previously mentioned interviews were asked to draw a sketch of a mobile wallet. As Linus Pauling once said: "The best way to get a good idea, is to get lots of ideas". Thus, the interviewees' ideas ended as sketches for four different wallets; one for each of the user groups. The sketches from each group were then mixed into one sketch, in what Pugh [19] calls controlled convergence. Pugh's approach is widely used, among others by Buxton [1]. Besides controlled convergence, which is about discarding ideas or part of ideas, Pugh used another notion, called concept generation.

Scenarios. Concept generation is about expanding the scope by adding new ideas. In this project, the new ideas came from the writing of scenarios that followed the sketching process. “The scenario is a narrative written in a natural language. It focuses on a user using the system. The goal of the scenario is to explore design solutions” [16]. According to Nielsen [16], personas and scenarios are inextricably linked, as personas are useless without scenarios. Below we present the mother scenario:

> It is afternoon, and Anita has just picked up Oliver and Casper from day-care and is now at the supermarket to buy groceries for supper. The children are hungry and troublesome, and therefore Anita wants to do the shopping quickly, so she can go home with the children and give them something to eat. At the self-service check-out desk she scans the goods, and when she has just opened up her mobile wallet to pay, Oliver starts begging for candy from the shelf right next to the self-service desk. This interrupts her in the act of paying, as she has to keep track him and ensure that he does not take any candy from the shelf. Meanwhile, more customers have lined up behind her. She notices this and hurries to pay. She chooses to pay cash, presses the ‘Pay’ button, types the PIN, and presses ‘OK’, and then hurries out of the supermarket with children and bags in her arms.

4.2 Development Phase

Based on the results from the Suggestion phase, four initial prototypes were created. One issue was to integrate all comments and remarks made by the participants, since there were differences within the four groups. However the aim was not to create a finite solution. Rather the aim was to get ideas from users to inspire the development of the prototypes.

For this we used mock-ups. A mock-up is often used as a topic for conversation in for example an interview, but the mock-ups in this project were used as prototypes. A prototype is a more or less functional model that enables stakeholders to interact with

Table 2. Initial prototypes used in the evaluation phase

<table>
<tr><th colspan="4">Prototypes</th></tr>
<tr><th>Young Teenager</th><th>Young Adult</th><th>Mother</th><th>Business Man</th></tr>
<tr>
<td>
</td>
<td>
</td>
<td>
</td>
<td>
</td>
</tr>
</table>

the imagined product. In that way, the prototype can be tested by the intended users in realistic environments, which leads to the designers' becoming aware of things, they had not thought of themselves. A prototype is a great help in the design process because of the fact that the designers are brought to completely new considerations, when they are going to take something from inside their minds and make it into something physical. The outcome of the development phase was four initial prototypes as illustrated in table 2. The prototypes are evaluated in the following phase.

4.3 Evaluation Phase

The Evaluation phase consists of an analysis of the gathered findings and an assessment of to what extent the artifact fills in the imperfections made explicit in the proposal from the Awareness of Problem phase. As Hevner et al. [8] explain it: "A design artifact is complete and effective when it satisfies the requirements and constraints of the problem it was meant to solve." Where the Evaluation phase focuses on what went good or badly, and decides whether or not iteration is needed, this section provides the basis for making these decisions. The questions asked were concerned with:

- The users' understanding of the mobile wallet's properties
- What impression they got when they first saw the mobile wallet
- What they thought about the properties that was specific for the mobile wallet compared to the physical wallet
- If they would like to have any other properties in the wallet
- Mentioning three things that they liked and three things that they disliked from the mobile wallet

Young Teenagers. Starting with the Young Teenagers, their first problem was caused by the 'Currency' tab in the top of the wallet. Both of the test users though it was nice to have, but said that they would not use it that much:

"Maybe it should not take up such a big part of the window, but I do not know where it should be placed instead, because you are not abroad that often […]. I would actually like if it converted automatically. And if you wanted to pay in Danish kroner, you should have the possibility to change it manually […]. Both [currencies] should be there so you do not have to calculate it yourself" (Young Teenager).

Another concern was regarding the security:

"I think that it is a bit insecure to have it in the phone. […] try to imagine that you lose your phone and other people can use it for travelling in your name" (Young Teenager).

One of the test users had an idea for how to load money to the mobile wallet:

"I think a lot about eBanking as I consider it almost the same as this phone-thing. I think that it should be possible to do it at home […] thus loading it from your computer at home into it [the mobile phone]" (Young Teenager).

Young Adults. The two Young Adults, who evaluated the first mock-up of their e-wallet, would not use the possibility of containing business cards in the e-wallet:

"[…] business cards do not belong [in a mobile wallet]. New mobile phones can hold so many data in the contacts/phonebook with emails, addresses, private phone numbers and work phone numbers and much more […] and so, it is completely needless" (Young Adult).

The icons in the e-wallet were preferred to the text buttons (see Figure 2), and it was suggested that the 'Receipts' button should be transferred into an icon:

"It looks rather unsystematic, careless [that the 'Receipts' button is placed where it is]. […] you could have an icon looking like a receipt" (Young Adult).

An application to the bank was also suggested:

"[…] I like having a receipt telling me how much money I have on my account. So, if there was an app to the bank […] like 'kontokik' [a function provided by the bank, enabling the user to see transactions on his account, but not to transfer money] for instance, where you could see how much money is on the account, what the money was spent at, and when withdrawals were made […] [it should be] just 'kontokik', not money transferring" (Young Adult).

Mothers. With regard to receipts, one of the test users from this group suggested that the balance should be a link leading to a kind of receipts or a list formed like a bank statement showing transactions. The other test person said that it would work for her if the receipts were in the e-wallet, as she did not need to have the receipts physically. She would, however, not keep all receipts:

"I would only keep those for expensive things. I would not keep those from buying milk and flour and eggs" (Mother).

The same person later proposed an additional function for the receipts:

"[...] if you keep the receipts, you could make some sort of fast search, to find out just how much money has been spent at the drug store this month [...] so, when you are working with your budgets or something like that, you could find out exactly what happens to the money" (Mother).

It was additionally suggested that there should be added radio buttons for different accounts, and that these accounts should have names:

"[…] it should be named as the account it came from, e.g. budget account. Because, my budget account is another account than my VISA card account. […] in that way I could see where they came from. My daughter has an account, [and it could] be there, and it could be that [card] you used for buying toys or clothes […]" (Mother).

Business Men. Both Business Men from this group liked the structure of the e-wallet, and both were strong advocates of sorting the cards in categories:

"They [i.e. the cards] should be placed underneath [a category] […]. Each page should only hold 7 – 9 – 10 [icons], just like when running a slide show; you should

only have five lines of text on a slide, or else people will not be able to take it in. So: categories and then moving downwards [to find other functions / cards]" (Business Man), and: "I think that [i.e. the categories] would be better. You could have some of the common [cards] up in the first [row], and furthermore have a section where you could make your own categories" (Business Man).

In addition to this, one of the test users suggested that the entire mobile wallet's contents shown on the front page should be ordered in categories represented as bars:

"[…] so when I entered [the wallet] I would like to have a bar called 'Payments', and when I pushed it some of this [pointing at the top row of cards] would appear. And afterwards some 'ID'. And then there could be a 'Other cards'. And then I might have the possibility of structuring it myself, if I would want an additional bar" (Business Man).

One of the test users moreover suggested the payment page as a picture of the receipt, which after the completed payment is sent to the mobile wallet. The payment page should therefore hold the name of the receiver of the money:

"[It should] show the hotel's receipt or the 'restaurant's receipt. That is, with name, so it can be used as receipt you can use as a voucher in your accounts" (Business Man).

4.4 Conclusion Phase

The test users for the second mock-up of the Young Teenager's mobile wallet, reflected upon how many items in the mobile wallet, they would like to secure by PIN or a password, in order to prevent strangers spending their money or using personal data, if the mobile phone should get lost or stolen. Both test users considered it insecure to have the passport in the mobile wallet, but disagreed on how many of the wallet's cards should be secured. The usability goal of safety has thus not been fully reached, as the mobile wallet that was tested not made the users fell completely safe. The revised prototypes after the evaluation phase are presented in table 3.

The tests of the second mock-up revealed that the usability goal of learnability has not been reached by this version of the Young Adult's mobile wallet, as both test users did not understand the 'Load' button. The test users additionally proposed a lot of new properties for the mobile wallet, e.g. a text appearing when payment is completed, a 'cancel' and a 'Load' button on the payment page, bus passes, a possibility to change the structure of the mobile wallets front page, a bank application, and many more. All these additional properties indicate that more utility is needed in the mobile wallet, and the usability goal of utility has therefore not been reached.

During the user tests of the second mock-up of the Mother's mobile wallet, one of the test users proposed a text appearing on the mobile wallet's screen when payment is completed. The adding of an eBanking function, a receipt option, and an automatic scanning of membership cards were proposed as well. A further suggestion was that the payment methods should include the user's different accounts, as a user might have more than one bank account. Moreover, the test users had different suggestions for what should happen to the receipts if a mobile wallet was introduced, and a further investigation of this matter is therefore needed. All these things lead to the conclusion that the usability goal of utility is far from being reached by this version of the mobile wallet.

Through the user tests of the second mock-up representing the Business Man's mobile wallet it became clear that a category structure was preferred to the structure with all the cards visible on the wallet's front page. The mobile wallet would therefore reach the usability goal of learnability to a greater extent, if the category structure were applied. An additional thing that would improve the learnability is the moving of the receipts from their present place, into the payment methods where the test users thought they belonged.

Table 3. Revised prototypes

<table>
<tr><th colspan="4">Prototypes</th></tr>
<tr><th>Young Teenager</th><th>Young Adult</th><th>Mother</th><th>Business Man</th></tr>
<tr>
<td>
</td>
<td>
</td>
<td>
</td>
<td>
</td>
</tr>
</table>

5 Discussion

The way the user tests of the mock-ups were conducted proved to be very useful for this project, as the interview approach to the tests, allowed for explanations when needed. And they were indeed needed. Some of users, had difficulties grasping the idea of mobile payments. Those who understood it had, on the other hand, many questions, especially concerning security and other aspects of mobile payments that are still uncertain. The user tests additionally revealed that it is of great importance, when testing an innovative product, to ask the test users to ignore the questions of whether they would use it, as this showed to affect a couple of the tests. Another observation showing that some users did not quite grasp the idea was made when some of the users suggested that the mobile wallet should hold the possibility of reading text messages and checking Facebook, because they would not want to be without it. Along the way, it was therefore decided to explain to the test users, that they still had all their other functions in the mobile phone, and that the mobile wallet was just another function.

The user tests did moreover inspire to asking further questions that had not been planned, and which might not have been asked to all the test users. But in the given situation, they seemed right to ask. For example, if the test users proposed ideas that had not been proposed before, it happened that the test users in the following tests were asked about this proposal, in order to have their opinion. This project was, however, an explorative design project, and nothing was given in advance. It was therefore all right to test several ideas. An expert is after all a person who has tested all kinds of solutions, in order to find the right one, and to learn from those who went badly.

Through the last iteration of user tests, several new ideas were proposed, and the user groups were still getting more inspired by each other's designs of the mobile wallet. The evaluation assessed that a new iteration is needed, it is concluded that the mobile wallet proposed by this project, is not yet ready to be launched. It was, however, never the purpose of this project to design a fully functional mobile wallet, but merely to broach the issue. This is achieved by proposing the set of properties for the mobile wallet. It is a possibility that the mobile wallet could hold a settings function allowing the user to edit the contents of the wallet, but it need to be explored if it is desirable for the users to have a lot of options.

6 Conclusion

A big challenge in the work with the mobile wallet was to clarify what functionalities each button should have. There were almost as many opinions as there were test users. This was revealed in the evaluations, where the usability goal of utility was the goal most far from being reached. The usability goal of learnability was not close to being reached either, mainly because the test users within each group had different opinions about the structure of the wallet and kept being inspired by the mock-ups of the other user groups' mobile wallets. It could thus have been tested whether it was possible to create one single mobile wallet for all users, by having one group's test users to test another group's mock-up. This would explore how it would work for them and what changes they would propose. By switching mock-ups through many iterations of tests, the possibility of having one design satisfying all the users, could thus have been tested. This project did, however, not has this approach, but focused on developing wallets for different user groups. It was from the beginning assumed that a standardized wallet would hold many customization options, hence confusing the user. This could lead to a situation where the user would not want to use the mobile wallet, and then we are back again where we started. The evaluation revealed that all the user groups actually had suggested such a settings function to be added to the wallet. It is therefore needed to be explored, how the users would use such a wallet, for instance through user tests of more functional prototypes allowing them to interact with the artifact representing the mobile wallet. The test users who suggested the adding of a settings function might, however, have done so, as they did not like to say directly that they wanted the mobile wallet to be different from what they were presented with.

Due to the small number of subjects, the findings of this project cannot be generalized to population. Instead, the findings can be generalized to theory and support further research of the emerging mobile payments.

Acknowledgments. The Danish Enterprise and Construction Authority grant number ERDFH-09-0026 sponsored this research.

References

1. Buxton, W.: Sketching User Experience: Getting the Design Right and the Right Design. Morgan Kaufmann, San Francisco (2007)
2. Dahlberg, T., Mallat, N.: Mobile Payment Service Development-Managerial Implications of Consumer Value Perceptions. Citeseer (2002)
3. Dahlberg, T., Mallat N., Ondrus J., Zmijewska A.: Mobile Payment Market and Research: Past, Present and Future. Presentation at Helsinki Mobility Roundtable, Helsinki, Finland (2006)
4. Dahlberg, T., Mallat, N., Ondrus, J., Zmijewska, A.: Past, Present and Future of Mobile Payments Research: A Literature Review. Electronic Commerce Research and Applications 7(2), 165–181 (2008)
5. Ferguson, N.: The Ascent of Money: A Financial History of the World. Penguin Books, London (2009)
6. Garcia-Swartz, D., Hahn, R., Layne-Farrar, A.: The Move toward a Cashless Society: A Closer Look at Payment Instrument Economics. Review of Network Economics 5(2), 175–197 (2006)
7. Hevner, A., Chatterjee, S.: Design Science Research in Information Systems. Design Research in Information Systems, 9–22 (2010)
8. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28(1), 75–105 (2004)
9. Hilavuo, S.: Business Evolution of Mobile Services. In: Managing Mobile Services-Technologies and Business Practices, Chichester, pp. 17–45 (2005)
10. Kim, H., DeVaney, S.A.: The Determinants of Outstanding Balances among Credit Card Revolvers. Financial Counseling and Planning 12(1), 67–77 (2001)
11. Knights, D., Noble, F., Vurdubakis, T., Willmott, H.: Electronic Cash and the Virtual Marketplace: Reflections on a Revolution Postponed. Organization 14(6), 747 (2007)
12. Li, H., Leckenby, J.D.: Examining the Effectiveness of Internet Advertising Formats. Internet Advertising: Theory and Research 203 (2007)
13. Mallat, N.: Exploring Consumer Adoption of Mobile Payments: A Qualitative Study. The Journal of Strategic Information Systems 16(4), 413–432 (2007)
14. Mallat, N., Rossi, M., Tuunainen, V.K.: Mobile Banking Services. Communications of the ACM 47(5), 42–46 (2004)
15. Munck, M.G. The Future Is Contactless. Presentation from a Get Fit Meeting at Nets on the 30th of November (2010), `http://www.cfir.dk/Forside/Arrangementer/GetFIT-Novemberthe30th/Slidesfraarrangementet.aspx`
16. Nielsen, L.: Engaging Personas and Narrative Scenarios. Phd Series, vol. 17. Samfundslitteratur, Copenhagen (2004)

17. Ondrus, J., Pigneur, Y.: Towards a Holistic Analysis of Mobile Payments: A Multiple Perspectives Approach. Electronic Commerce Research and Applications 5(3), 246–257 (2006)
18. Peffers, K., Tuunanen, T., Gengler, C.E., Rossi, M., Hui, W., Virtanen, V., Bragge, J.: The Design Science Research Process: A Model for Producing and Presenting Information Systems Research (2006)
19. Pugh, S.: Total Design: Integrated Methods for Successful Product Engineering. Addison-Wesley, Reading (1991)
20. Sharp, H., Rogers, Y., Preece, J.: Interaction Design: Beyond Human-Computer Interaction, 2nd edn. Hoboken, Chichester (2007)
21. Takeda, H., Veerkamp, P., Yoshikawa, H.: Modeling Design Process. AI Magazine 11(4), 37 (1990)
22. Vaishnavi, V., Kuechler, W.: Design Science Research Methods and Patterns: Innovating Information and Communication Technology. Auerbach Publications, Boca Raton (2008)
23. Van Aken, J.E.: Management Research as a Design Science: Articulating the Research Products of Mode 2 Knowledge Production in Management. British Journal of Management 16(1), 19–36 (2005)
24. Wells, W.D., Gubar, G.: Life Cycle Concept in Marketing Research. Journal of Marketing Research 3(4), 355–363 (1966)
25. Zheng, X., Chen, D.: Study of Mobile Payments System. In: IEEE International Conference on E-Commerce Technology, CEC 2003 (2003)

A Fitness-Utility Model for Design Science Research

T. Grandon Gill and Alan R. Hevner

Information Systems and Decision Sciences, College of Business,
University of South Florida, Tampa, FL 33620
{grandon,ahevner}@usf.edu

Abstract. Current thinking in design science research (DSR) defines the usefulness of the design artifact in a relevant problem environment as the primary research goal. Here we propose a complementary evaluation model for DSR. Drawing from evolutionary economics, we define a fitness-utility model that better captures the evolutionary nature of design improvements and the essential DSR nature of searching for a satisfactory design across a fitness landscape. We conclude with a discussion of the strengths and challenges of the fitness-utility model for performing rigorous DSR.

Keywords: Design science research, design evaluation, usefulness, utility, fitness, evolutionary economics.

1 The Dependent Variable in Design Science Research

Current thinking in design science research (DSR) defines *utility* as the primary research goal (e.g. [1, p. 80]). In this context, the close relationship of utility to practical *usefulness* is emphasized. The choice of usefulness as the pre-eminent dependent variable for DSR ties it to earlier MIS research exploring appropriate dependent variables for information systems [2, 3]. It also establishes a clear relationship between DSR and the influential technology acceptance model (TAM) for information systems, where usefulness plays a pivotal role in motivating use [4]. Given these strong connections to existing well established research streams, does it even make sense to question if usefulness should *always* be our central criteria for evaluating design?

Being contrarians, we do feel the search for the dependent variable in DSR requires some rethinking. Here, we consider a pair of alternative dependent variables: *design fitness* and *design utility*. In the case of fitness, we particularly focus on its biological meaning—the ability of an entity to reproduce itself and evolve from generation to generation. In the case of utility, rather than viewing it as being roughly equivalent to usefulness, we focus on its meaning in fields such as economics and decision sciences, where it serves as the basis for ranking decision alternatives. Naturally, usefulness plays an important role in determining both fitness and utility. Neither of these variables, however, is solely determined by usefulness. Indeed, we believe that understanding the relationship between the three variables via a new fitness-utility model complements current thinking and provides important insights into the nature of design science.

H. Jain, A.P. Sinha, and P. Vitharana (Eds.): DESRIST 2011, LNCS 6629, pp. 237–252, 2011.

We begin by clarifying the frequently misunderstood concept of the *design artifact*. We then explore the nature of our two proposed dependent variables, fitness and utility, as they are defined in biology, economics, and in the emerging interdisciplinary field of evolutionary economics. Subsequently, we consider how these concepts can be employed in the context of artifacts and designs. We then consider how the guidelines of design science research may be better understood in the context of the fitness-utility model. Finally, the specific benefits and challenges of applying the fitness-utility model for DSR are discussed.

2 Design Artifacts

Central to the notion of DSR is the concept of a design artifact. IT artifacts are broadly defined as constructs (vocabulary and symbols), models (abstractions and representations), methods (algorithms and practices), and instantiations (implemented and prototype systems). [1, p. 77] More generally, artifacts can be viewed as the symbolic representation or physical instantiation of design concepts. Even within a discipline such as MIS, they are not necessarily limited to information systems. Rather, MIS artifacts include organizational designs, process designs, and other intentionally constructed entities relating to information systems.

Conceptually, we can view the design process as a series of layers, as seen in Figure 1. The top layer, the design space, can be viewed as the collection of all possible designs and requirements. Obviously, its contents "exist" in abstract terms only since such a complex design space is infinite. Conceptually, then, we can imagine that the space is partitioned between a few known and many unknown designs. The design process begins with a search of this space in order to identify a particular position, which can be referred to as a *design candidate*.

Between the design space and use artifact layers we find the design artifact layers, of particular significance to DSR. Once a design space candidate has been chosen, we can begin to develop artifacts. As previously noted, these may be symbolic or physical representations of our selected location in the design space. These artifacts may serve a variety of purposes:

1. Providing evidence of design feasibility - Can the proposed design be implemented and does the proposed design meet the requirements? Building feasibility artifacts moves designs across the unknown/known partition.
2. Providing evidence of the value of the design - Does the design offer benefits unmatched by competing design candidates? Here the objective becomes to establish an ordinal valuation that can be used to rank candidate designs.
3. Determining the most effective representation of the design – How can we best communicate the intricacies of the design to the implementators (e.g. architects, programmers).
4. Constructing the actual use artifacts - A blueprint is a construction artifact that serves to guide the physical construction of a house; source code is a construction artifact that serves to generate the programs that are distributed to users.

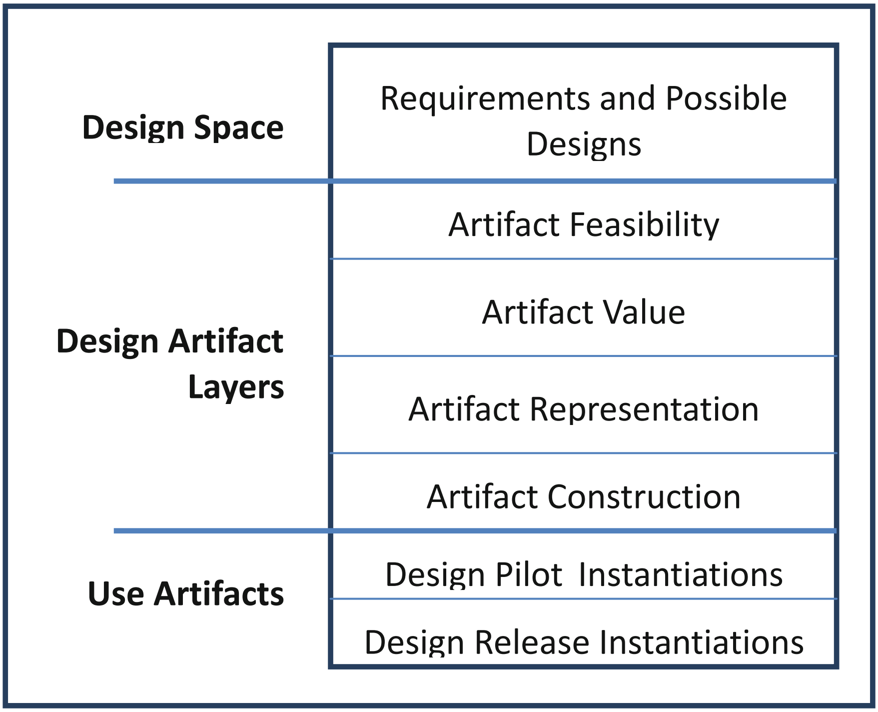


Fig. 1. Design Artifact Abstract Layers

The use artifacts are divided between pilot test instances—for which returning to the design cycle is intentionally left open as a possibility—and release use instances, for which further redesign is not anticipated. While this conceptual scheme obviously maps directly to IT artifacts such as software, it should be recognized that organizations frequently employ a phased roll out of non-technology artifacts, such as organizational structures or incentive plans, with the same notion that the design may later be tuned based upon early experience.

The particular significance of design artifact layers to DSR stems from their nature. As noted previously, the design space itself is too amorphous to be investigated directly. We need a physical representation or symbolic description of a particular design candidate—in other words, an artifact—if we are to conduct meaningful research. The investigation of use artifacts, on the other hand, is largely the domain of behavioral research. Inasmuch as they have already been constructed, the principles incorporated in their design are likely to be of less interest than the principles determining how their use impacts the entities (e.g. organizations) in which they are embedded. Nevertheless, it is certainly possible—indeed probable—that important principles that may guide future design can be acquired by observing constructed instances in use. This highlights the complementarity and need for communication between design science and other research paradigms.

3 Fitness and Utility

Based on our understanding of the layers of the design artifact, we can now move to an exploration of how better to understand and evaluate the artifact in DSR. Two concepts from other disciplines for this task are fitness (biology) and utility (economics).

3.1 Fitness

To understand fitness, it is useful to begin by proposing two alternative definitions of the fitness of an organism:

Fitness Definition #1: The fitness of an organism describes its ability to survive at a high level of capacity over time.

Fitness Definition #2: The fitness of an organism describes its ability to replicate and evolve over successive generations.

Which definition of fitness you prefer likely depends on your perspective. If the individual in mind were our personal physician, we would strongly prefer he or she focus on definition #1. Terms such as physical fitness, mental fitness and emotional fitness all correspond to this general class of definition. If, however, the individual were an evolutionary biologist, definition #2 would be overwhelmingly preferred. An organism lacking the capacity to reproduce and evolve rapidly goes extinct. What is important about the distinction between definition #1 and definition #2 is that their outcomes are not necessarily correlated. This is graphically illustrated by the experience with human populations, as discussed in Example 1.

Example 1: Two Versions of Fitness in Populations

At the end of the 18th century, Thomas Malthus proposed that any increases in the individual fitness (definition #1) of human populations would lead to a rapid increase in reproductive rate (a contributor to definition #2) that would quickly erase the gains in individual fitness and would, in the long run, reduce individual fitness since gains in food supplies tended to be arithmetic whereas changes in reproductive rates tended to be geometric [5, p. 6]. What has actually happened, however, is in stark contrast to predictions. After a period of adjustment, as individual fitness increases, evolutionary fitness (definition #2) has actually declined.

To illustrate this phenomenon, it is useful to consider two measures: life expectancy (a proxy for definition #1) and fertility rate (a proxy for definition #2). In an organism that employs sexual reproduction, fertility rate represents the number of children each female of the species produces over her lifetime. In human populations—where the number of male babies is slightly higher than the number of female babies—a stable population requires a value is slightly over 2. In much of the industrialized world, this value has fallen far below that stable value. For example, the 2006 U.N. Economic and Social Affairs agency estimated Japan's 2000-2005 fertility rate at a shockingly low 1.29. During the same period, the U.S. had an estimated value of 2.04. Based on definition #1, the fact that Japan has the highest life expectancy in the world among major industrialized nations would imply high fitness. With respect to definition #2, on the other hand, such low birth rates suggest a population that is decidedly unfit from an evolutionary standpoint.

We will henceforth *always* refer to the second definition of fitness when we use the unqualified term. There are two reasons for this. The first is that the population-focused view of fitness is generally more sensible when long term systems, such as information systems, are studied. Second, as we shall see later, it would relatively easy to treat variables such as system use or usefulness as a proxy for fitness according to definition #1, implying that little benefit is likely to be derived from advocating definition #1 in place of currently popular dependent variables. Because of the tension between definitions #1 and #2, already noted in the population example, we would expect that examining definition #2 might offer new insights.

Prior to leaving the subject of fitness, it is useful to introduce a model used by evolutionary biologists, that of the fitness landscape. Such a landscape represents a functional relationship between individual attributes, such as specific genes, and the fitness of an organism. For example, if an organism's fitness were determined by N attributes, x_1 through x_N, its fitness landscape would be described as: Fitness = $f(x_1, x_2, \ldots, x_N)$. Fitness landscapes change over time as a result of forces such as the organisms' collective impact on the environment, the impact of co-evolution of other organisms, and the impact of unpredicted events that occur entirely outside of the systems being studied, popularly referred to as black swans [6].

3.2 Utility

Similar to fitness, the term utility is used in a number of ways. When we consider the utility of a tool, we are normally referring to its usefulness. As currently used in the context of DSR, that is the prevailing meaning. Hevner, et al. [1, p. 83] state: "The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods." This implies utility to be a characteristic of the design and its intended application context.

Economists, on the other hand, employ the term utility in a different way. Specifically, they posit each individual to have a utility function that can be used to rank choices in the context of decision-making. The assumption that individuals seek to maximize utility is, in fact, foundational to the field of economics. In early economic theory, the assumption was made that utility was determined by current consumption. More recently, however, it has been generally recognized that many factors contribute to economic utility beyond direct consumption, such as relative income, expectations, social context, and goals [7].

To distinguish between the two usages of the term utility, we will refer to the first as *usefulness*. In this context, we apply the broadest meaning of the term—including factors such as efficacy in performing the task (including performance), range of task cases performed, ease of use, ease of learning, and cost-benefit in the performance of a task. Essentially, we assume that any artifact characteristic that impacts task performance directly can be classified under the usefulness category. Presumably, if our choice of a tool was dictated strictly by usefulness, as just defined, then there would be little reason to distinguish between the two meanings of utility. When we employ the term utility in the rest of the paper, however, we assume its economic meaning and further assume that it represents a complex function that is not adequately described by the single usefulness dimension.

3.3 Evolutionary Economics

Evolutionary economics is a field that examines economic systems from the perspective of evolution. As it happens, the foundational assumption of the field ties the notions of fitness and utility together. At the risk of oversimplifying, the basic concept that drives the field is that, as humans, our utility function has evolved as a response to the fitness landscapes we have faced and, as a consequence, is tuned towards maximizing fitness. The rationale is stated as follows by Gandolfi et al. [8, p. 97]: "Given the logic of natural selection, it is difficult to conceive how, for any living entity, a preference for maximizing fitness could fail to evolve."

The argument is based upon the fact that, on a static landscape, high fitness individuals will tend to crowd out lower fitness individuals. Because fitness landscapes are themselves subject to change (as noted previously), traits that promote diversity—e.g. an urge, on the part of some individuals, to seek out new peaks—are also likely to survive over time in some percentage of a population, described as *evolutionarily stable strategies* (ESS). The percentage of a population described by a particular niche ESS may grow after sudden shifts in fitness that increase in the value of the strategy, while niche strategies may well decline in percentage during long periods of stability in which highly visible high fitness peaks draw an increasing portion of the population. Regardless of where an entity exists on the fitness landscape, however, utility will tend to drive it toward local peaks.

A sensible argument can be made that our current utility preferences do not map well to fitness. Numerous researchers have demonstrated that we, as human beings, are far from rational in our processes of choice [9]. There are a number of ways to respond to this. First, evolution is slow and—particularly over the past 250 years—changes in the environments we face as a consequence of the industrial and information ages have been so rapid that it would be inconceivable that our utility preferences could have kept up. Fortunately, our built-in genetic utility function also imbues most of us with a desire to learn and, as a consequence, our utility function can adapt to our changing environment through that mechanism, as opposed to natural selection. Second, it is actually very rare that we encounter tasks with such well-defined inputs (e.g. where probabilities are fully known) in our day-to-day life. Indeed, it is often the case that when we attempt to quantify such values in order to make our decision making more precise, we are in fact deluding ourselves. Third, even individuals who have done extensive research into the "irrationality" of our decision rules acknowledge that there are many contexts where these decision rules prove to be beneficial [10].

The key point here is that utility can be treated as the mechanism by which we make choices when confronted with a fitness landscape. Obviously, it is not perfect. Rather, it represents an estimate-of-fitness that we can apply to make decisions presented by such a landscape. With this principle in mind, we turn to its specific application in the domain of DSR.

4 Relationship of Fitness, Utility, and Usefulness in Design Science

The concepts of fitness and utility can readily be applied to the design of systems. If we revisit Figure 1, it should be evident that the design space is an example of a fitness landscape, with each design candidate being an entity that can be located on that landscape. Design artifacts perform two key roles in the design search process:

1. They provide evidence that a particular design candidate is feasible, has value, can be effectively represented, and can be built. This serves to help us better understand the shape of the design fitness landscape, moving combinations from the unknown to the known category.
2. Through careful evaluation, they provide a basis for choosing between alternative designs.

The first of these directly impacts our knowledge of design fitness. The second refines our estimate-of-fitness that is a basis for choice; it therefore involves changing our utility function through learning.

Where design systems differ from biological evolution is in the role played by intentionality. The mechanisms of evolutionary change—such as production of new gene combinations through sexual reproduction and mutation—are posited to exert their influence with considerable randomness. While survival rates serve to cull the low fitness organisms from the population, the actual construction of such organisms is unguided. In the design space, on the other hand, designers intentionally concentrate on areas of the design fitness landscape where promising candidates have been identified. What that means is that while utility serves as an estimate-of-fitness for design artifacts, it also feeds back into the fitness landscape itself since a low fitness evaluation for a particular design candidate will discourage further investigations into nearby regions of the design landscape. This, in turn, reduces the fitness of those regions since placing less effort into building artifacts based on a particular design will necessarily reduce the flow of future artifacts based on that design (which is how we define fitness). Moreover, the shape of the utility function is likely to be guided by two forces: the nature of the evaluation artifacts being studied *and* by actual experience from artifacts developed for use. Thus, the experience of artifacts placed in practice has the ability to impact the design fitness landscape just as evaluation artifacts do. Thus the new fitness-utility model can re-frame DSR as follows:

The goal of DSR is to impact the design space so as to ensure a continuous flow of high fitness design artifacts. This impact is accomplished in two ways: through the production of artifacts that demonstrate the feasibility of new designs and through improving the utility function that we use to assess the fitness of evaluation artifacts.

This definition, of course, represents a type of artifact, one where fitness and utility replace potential usefulness as dependent variables. What we shall now do is to identify ways in which this approach differs from our prior understanding of DSR. We do this by examining the seven DSR guidelines proposed by Hevner et al. [1].

5 Fitness and Utility Goals in DSR Guidelines

How does the fitness-utility approach to DSR differ from the existing paradigm? In this section we concentrate on how fitness defined in terms of reproductive efficacy and utility defined in terms of a choice frontier alter our perspective on DSR.

5.1 Guideline #1: Design as an Artifact

The fact that DSR is constrained to deal with the concrete by this guideline is important in distinguishing it from behavioral research. Moreover, the original guidelines are sufficiently broad in their definition that meta-design findings (e.g. a list of attributes that contribute to design quality) would, itself, constitute an artifact and would therefore—quite rightly—fall under the DSR heading. Under the fitness-utility approach, the term "produce" would be too limiting, however, since research leading to changes to the design utility function would fall under the approach. Thus, research can radically change the design space without necessarily producing a design artifact. A radical restatement of this guideline would be as follows:

Guideline #1: The objective of the fitness-utility model of DSR is to impact the design space through the creation and evaluation of design artifacts.

5.2 Guideline #2: Problem Relevance

While we agree with the continued importance of relevance, the problem with the existing statement of guideline #2 is that, from a practical standpoint, it tends to constrain the time horizons for design research. We often cannot foresee what problems will be relevant for the future of IT. The challenge this unpredictability presents to DSR is that if you try to anticipate the important long term problems that a design will solve, it will be nearly impossible to get them right. Thus, being overly problem-focused demands a shorter term outlook. Another way of looking at the issue is to use the analogy of constructing a puzzle. At the beginning of a puzzle, as in a design process, you have a collection of pieces that can only be put together in certain ways. True "problems"—in the form of missing pieces--tend to be discovered near the end of the assembly, when the gap is identified. If we require that our design science solve an important problem, we may need to wait until we know what is missing. With this caveat, we do not recommend any changes to DSR Guideline #2.

Guideline #2: The objective of DSR is to develop technology-based solutions to important and relevant business problems.

5.3 Guideline #3: Design Evaluation

The fitness-utility model recognizes a large number of characteristics that could potentially be used to assess design fitness. These are illustrated in Figure 2. The area within the fitness ellipse outside of the intersection with the usefulness ellipse reflects characteristics that can impact fitness that are not a direct a result of usefulness (although they may be correlated with it). Those characteristics listed in Figure 2 are intended to serve as an incomplete list of examples that will now be discussed. We begin, however, by revisiting usefulness.

As illustrated in Figure 2, the potential usefulness of a design artifact still plays a key role in assessing fitness, as it did in the original model. What the figure also suggests, however, is that there may be times when a design artifact becomes so useful that it actually inhibits further improved designs—much the way increased life expectancy (i.e. fitness definition #1) has become associated with below replacement fertility rates (i.e. fitness definition #2). In fact, the tendency of organizations to stick with designs that have proven useful is a well-documented phenomenon known as the Innovator's Dilemma [11].

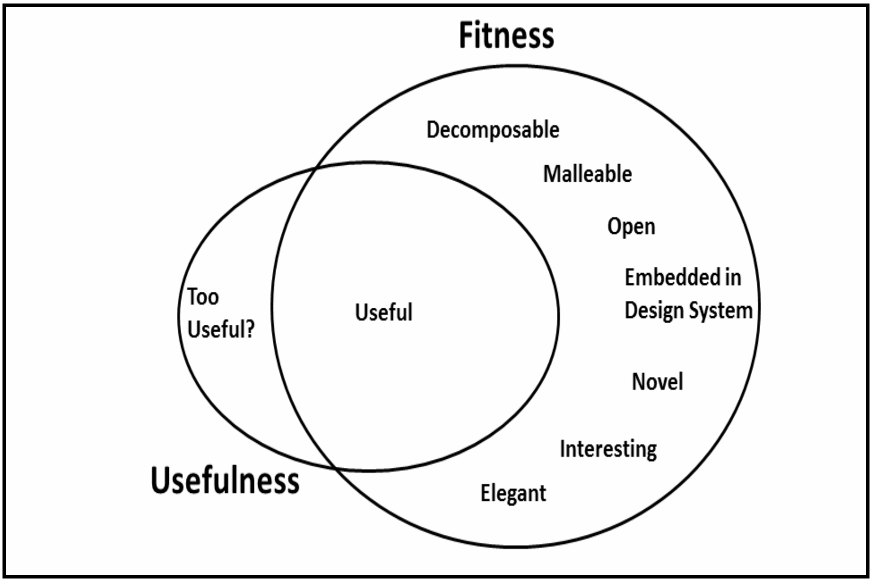


Fig. 2. Design Candidate Fitness Characteristics and Usefulness

The other key characteristics of artifact fitness are briefly discussed here:

Decomposable. The seminal work that launched the study of design science is Herbert Simon's *The Sciences of the Artificial* [12]. The second half of the book is largely devoted to explaining why systems tend to evolve from nearly decomposable subsystems. Indeed, even under the existing design science goals, decomposability is likely to exert a strong influence on design quality and would therefore be evaluated as part of the design. In addition, such systems tend to be easier to construct, since work on individual components can be conducted separately. The particular difference that the fitness-utility approach would engender involves the reproduction and evolution of partial designs. Where a design cannot be decomposed into nearly independent subsystems, evolution of the design would tend to be a matter of all-or-nothing. Where a design is built upon separable systems or constructions, on the other hand, pieces of the design—strands of *design DNA* to use a biological analogy—may exhibit high fitness and evolve rapidly while others may remain static or be discarded.

Malleable. Related to decomposability, the malleability of an artifact represents the degree to which it can be adapted by its users and respond to changing use/market environments [13, 14]. MIS research has demonstrated that users frequently employ tools for unintended purposes. We would expect that such adaptation would allow designers to evolve artifacts to support these uses more effectively.

Open. Another characteristic that has the potential to impact design fitness is the degree to which artifacts are open to inspection, modification, and reuse. Openness tends to encourage design evolution by making it easier both to see how an artifact is designed and to modify existing components of the artifact. For example, an information system created as an open source application has a significant advantage over a proprietary design in terms of its ability to evolve rapidly.

Embedded in a Design System. We would expect design artifacts that are the product of a sustainable design system environment to evolve more rapidly than artifacts that are produced in a context where design is an unusual activity. This particular source of fitness can sometimes act as a counterweight to openness, as organizations with highly effective research and development activities may be reluctant to open up their designs and may use legal measures—such as patents and copyrights—to discourage unauthorized parties from evolving the original designs. An effective design system can produce a stream of design artifacts, however, even without the financial rewards that comes from transforming these into use artifacts.

Novelty. A design may be considered novel if it originates from an entirely new region of the design space. Once such a design candidate has proven viable, other design candidates from the same region are likely to follow in an attempt to locate the local peak on the fitness landscape. A particular challenge that novel design artifacts present is that the creative process through which they are envisioned may not meet the criterion of rigor suggested by the original guideline and the potential benefits of the design may be hard to evaluate.

Interesting. Normally, a design artifact is created in order to explore or demonstrate some specific purpose. From time-to-time, however, an artifact may demonstrate unexpected emergent behaviors that are worthy of subsequent investigation and the creation of subsequent artifacts. Social scientists (e.g. [15]) have long asserted that research which largely conforms to existing expectations yet also incorporates an unexpected element is most likely to interest other researchers.

Elegant. In many areas of design, such as architecture, consumer products and apparel, there is an ongoing tension described as form versus function. Function relates to practical usefulness. Form, in contrast, describes aesthetic elements such as appearance that do not necessarily serve a useful purpose, yet nevertheless increase the user's utility. The characteristic of an MIS design artifact that corresponds to form might best be referred to as *elegance*. Like quality, elegance is hard to define in a rigorous manner and yet characteristics that might be associated with it—such as compactness, simplicity, transparency of use, transparency of behavior, clarity of representation—can all lead to designs that invite surprise, delight, imitation, and enhancement.

If the fitness-utility approach is taken to DSR, then the evaluation criteria are where the utility function is to be shaped. Further thinking and research are needed in order to propose methods for formulating the utility model in specific DSR projects. Thus, we would require a restatement of the original design evaluation guidelines along the following lines:

Guideline #3: The fitness of a design artifact must be estimated using a utility function that considers the full range of characteristics that can impact the likelihood that the artifact will further be reproduced and evolve.

5.4 Guideline #4: Research Contribution

With respect to this guideline, the fitness-utility approach and the original approach are relatively similar. As originally stated, however, it is not clear that research that leads to better understanding of utility (i.e. estimating the fitness of a design artifact) would be included under the design heading. For this reason, a preferable rewording might be:

Guideline #4: Effective DSR impacts the design space through contributions in the area of the design artifact, design fitness, design foundations and theories, and/or design methods.

5.5 Guideline #5: Research Rigor

A particular challenge associated with the use of the term rigor is that it is perceived to be generally "understood" but is rarely defined. One definition that has been proposed (e.g. [7]) treats research rigor as consisting of three related elements: 1) the investigation is systematic, 2) a thoughtful balance is struck between the risk of accepting that which is false (Type 1 error) and rejecting that which is true (Type 2 error), and 3) challenging questions are posed. By this definition, the current guideline would tend to place considerable obstacles in the way of early stage design artifacts, inasmuch as: i) systematic search of the design space is generally impossible, ii) current standards of empirical research in the social sciences tend to lean heavily towards avoiding Type 1 error [16] making rejection of novel ideas more likely, iii) early stage design artifacts often leave challenging questions—such as scalability and relative benefits compared to alternative designs—largely unanswered. Rather than abandoning rigor altogether, the guideline could be revised as follows:

Guideline #5: DSR requires that the construction and evaluation of design artifacts be investigated employing a level of rigor appropriate to the nature and stage of design.

5.6 Guideline #6: Design as a Search Process

There is little need to change the spirit of this guideline, which captures perfectly the process of search in a fitness landscape. A slight modification to the wording is desirable, since the fitness-utility model assumes we are searching for high fitness artifacts in a design space.

Guideline #6: The search for high fitness design candidates and artifacts requires utilizing available means to reach desired ends while satisfying laws in the design space.

5.7 Guideline #7: Communication of Research

This guideline once again illustrates the preference for late-stage design research in the original conception of DSR. Management-oriented audiences, in particular, are unlikely to be impressed by designs whose usefulness has not been demonstrated. The fitness-utility approach would take an entirely different perspective. Where the goal is to exert impact on the design space (which is a fitness landscape), what makes sense is to target those communities most likely to initiate the next iteration of the design process through supplying resources, which would naturally include time, intellectual effort, facilities, and money.

Guideline #7: Design research must be communicated to those communities most likely to supply the resources required for future design using communication channels appropriate to each community.

6 Discussion – Pros and Cons of the Fitness-Utility Model

With its focus on reproductive fitness (i.e. definition #2) rather than individual artifact fitness (i.e. definition #1), the fitness-utility model offers both strengths and weaknesses when contrasted with the existing DSR paradigm, which is why we view it as a complement rather than as a competitor to the existing approach. We discuss five advantages of the new model followed by three challenges.

6.1 Makes the Researcher an Active Participant in the Design System

Because developers often publish research relating to the artifacts they are creating, a great deal of design research in IT is already action research. Under the fitness-utility model, however, even the non-technical researcher strives to play an active role in the design system through impacting fitness values in the design space. Successful research will, as a matter of definition, lead to either an increase or decrease in the production of new artifacts based upon the specific design candidate or candidates investigated. The fitness-utility model would also be predicted to maximize the potential impact of individual research contributions by focusing on early stage design. Thus, if the researcher's goal is to impact the design space, consistent with the goals of the fitness-utility model, the earlier the artifacts evolving from a particular design candidate can be identified, the better.

6.2 Provides an Alternative Basis for Evaluating Research Impact

Today, within the MIS research discipline, the impact of research is generally measured through the estimated quality of the publication outlet and through subsequent citations by other researchers. The fitness-utility approach offers another alternative: chart the evolution of subsequent artifacts contrasted with the findings of the research. If the artifact continues to evolve and incorporate design DNA deemed favorable by the research, then impact—in the truest sense of the word—has been achieved. The same can be said of research that stifles the further evolution of design DNA deemed detrimental to fitness. For example, if particular design practice (e.g. allowing the user to enter free form text into a textbox that is then used to query a database) leads to a security threat (e.g. malevolent SQL injection), impactful DSR that identifies this as a low fitness practice should reduce the frequency of the occurrence in later artifacts.

6.3 Aligns with Dynamic Environments

A central premise of this paper is that over time the evolutionary fitness of design artifacts becomes far more interesting than the use fitness of a particular artifact. The validity of this premise is likely to depend on the environment in which it is situated. For very static environments, for example, a particular use artifact may exist for a very long time. In such a world, the use fitness of the artifact is a matter of considerable interest. In a highly dynamic environment, on the other hand, the artifact's potential to evolve needs to be given much greater weight. Our belief is that such dynamism describes most environments facing IT designers today, and that forces such as globalization, social media, and advances in telecommunications will likely serve to increase environmental turbulence.

6.4 Recognizes the Inherent Limitations of Intended Usefulness

Our research suggests that while usefulness is likely to be the best single predictor of artifact use (a finding consistent with most TAM research), it is not a necessarily a very good predictor when applied by itself. In fact, a reasonable argument can be made that many of the most *interesting* (see [15]) findings of MIS revolve around examples where an IT artifact's impact was far different from the designer's intended use.

6.5 Encourages Collaboration between MIS Researchers and Designers in Other Fields

The fitness-utility approach specifically targets clients in the design communities supplying the resources necessary for further design evolution. In early stage IT design research these communities will likely contain a preponderance of researchers in technical fields such as computer science including many academics. Thus, we will have a strong incentive to collaborate with these communities if we are to exert impact. Where we may be able to contribute is in our understanding of the potential unintended consequences of artifacts employed in an organizational setting, as previously described. Having observed these consequences in the field and studied them in our literature, we are in a unique position to provide perspective to designers who may otherwise become overly focused on intended use.

6.6 Current Research Standards Do Not Reward Design Impact

Given that researcher rewards, including promotion and tenure, tend to be closely tied to measured research impact based on numbers of quality publications and citations, the fact that the fitness-utility model offers another approach to measuring impact over time—tracing how artifact design DNA changes as a consequence of research findings—may not be appealing to academic researchers. In the absence of institutional change with respect to how impact is defined, it may be hard for the fitness-utility model to gain traction.

6.7 The Framework for Evaluating Design Fitness Is Not Well Researched

Earlier in the paper, we proposed a number of non-use characteristics (Figure 2) that seemed likely to impact design fitness. This list was largely inducted from examples and could in no way be considered complete, rigorously derived, or rigorously supported. Unfortunately, there is little research into the characteristics that provide good estimates of design fitness as we have defined it. Stated another way, our design utility function is largely unexplored. This naturally presents a substantial obstacle to any research that attempts to estimate the fitness of a particular artifact. The largely unexplored forces driving fitness and utility are in stark contrast to the much better established approaches to evaluating design usefulness. Although the field laments its lack of theoretical base, constructs, and generalizability [1, p. 99], it has a plethora of these when contrasted with the fitness-utility model. Add to this the fact that immediate usefulness is likely to seem a more concrete research objective than fitness, and the researcher is likely to have a much easier time designing research under the existing paradigm.

6.8 Building Rigor for Fitness-Utility Research Requires Alternative Research Methods

It may be argued that the last challenge actually represents a considerable opportunity for future research into the factors that lead to fitness. Such research, however, is likely to substantially differ in character from the main body of existing MIS research. To understand fitness, you need to look backward in time in order to trace the evolution of an artifact. Indeed, it may take years to validate the actual fitness of an artifact—a necessary step if the characteristics contributing to fitness are to be identified. Thus, historical research methods are likely to play a much greater role that is the case in most contemporary MIS research (see [17]). In addition, fitness landscapes in general tend to be rugged, meaning that interdependencies between variables prevent decomposability. Such ruggedness can confound traditional statistical techniques. What this means is that data analysis techniques most preferred by MIS researchers may prove largely inapplicable in the analysis of sources of fitness.

7 Conclusions

Several times in this paper, we have posited that the fitness-utility model for design science research is better viewed as a complement to the existing usefulness model, rather than as a competitor. As illustrated in Table 1, which summarizes the analyses presented in this paper, the two models focus on different objectives, are most applicable to different artifacts, tend to examine different units of analysis, are appropriate for different time horizons, are likely to employ different research methods, and will tend to be of greatest interest to different client constituencies. We have already noted that high levels of usefulness may actually inhibit artifact

Table 1. Summary of Usefulness and Fitness-Utility Models

Characteristic	Usefulness Model	Fitness-Utility Model
Focus	Useful artifacts	Artifact reproduction and evolution (fitness) and the choice mechanisms guiding artifact design (utility)
Applicable artifacts	Construction and use	Feasibility and evaluation
Unit of study	Entire artifact	"Design DNA" within artifact
Time horizons	Short and medium-term	Long term
Source of rigor	Careful evaluation of intended use and expected performance	Systematic evaluation of non-usefulness factors that may contribute to fitness and the potential for unintended consequences
Most likely external (non-MIS research) clients	Developers and use clients	Researchers outside of MIS and R&D clients
Source of models	Study of current artifacts in the field	Study of historical progression of artifacts based upon a particular design candidate
Particular value offered by MIS research	Understanding the organizational context in which artifact development and use takes place	Understanding the role played by unintended consequences in typical artifact implementation; broad perspective on factors that influence artifact success
Desired impact of research	Improved design and development of useful artifacts and better understanding of the factors that make an artifact useful	Improving fitness of desirable design DNA and suppression of undesirable strands; better understanding of the factors that increase real-world artifact fitness leading to improved choice between alternative design candidates.

evolution. There is likely to be quite a bit of causality here—an organization making a large investment in designing and deploying a use artifact is unlikely to view the tendency to evolve rapidly as a major benefit. To the contrary, such a manager is most likely to appreciate an artifact that is highly useful and is likely to remain that way as long as possible. For that clientele, maximizing the fitness of design, as we have defined it, is more likely to be more scary than desirable. Moreover, an understanding of the factors contributing to usefulness is central to the fitness-utility model. Many factors outside of usefulness may contribute to fitness, but we expect usefulness will typically prove to be the single most important factor in most design settings.

Nevertheless, we believe the fitness-utility model for DSR is too important to ignore. It is our strong belief that an artifact that continues to evolve will *always* end up outperforming an artifact that fails to evolve, regardless of their respective usefulness at the time they were conceived. This is the core of Christensen's innovator's dilemma [11] and if we do not recognize this process, we are ignoring a major force that shapes today's competitive environment. As we have pointed out, however, such research is likely to adhere to different guidelines (Section 5) and depart considerably from existing DSR practices. It is our goal in this paper to alert researchers and reviewers of these differences and offer some justification as to why they are necessary. In doing so, it is our hope to stimulate future DSR thinking along the lines of the fitness-utility model. We hope to advance these ideas further by describing case studies and performing DSR projects in which evaluation is based on the fitness-utility model.

Acknowledgements. The authors are grateful for the insightful comments by the anonymous reviewers that led to improvements in the paper.

References

1. Hevner, A., March, S., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28(1), 75–106 (2004)
2. DeLone, W., McLean, E.: Information Systems Success: The Quest for the Dependent Variable. Information Systems Research 3(1), 60–95 (1992)
3. DeLone, W., McLean, E.: The DeLone and McLean Model of Information Systems Success: A Ten-year Update. Journal of Management Information Systems 19(4), 9–30 (2003)
4. Venkatesh, V., Davis, F.: A Theoretical Extension of the Technology Acceptance Model: Four Longitudinal Field Studies. Management Science 46(2), 186–204 (2000)
5. Gill, R., Glazer, N., Thernstrom, S.: Our Changing Population. Prentice Hall, Englewood Cliffs (1992)
6. Taleb, N.: The Black Swan: The Impact of the Highly Improbable. Random House, New York (2007)
7. Gill, T.: Informing Business: Research and Education on a Rugged Landscape. Informing Science Press, Santa Rosa (2010)
8. Gandolfi, A.E., Gandolfi, A.S., Barash, D.: Economics as an Evolutionary Science: From Utility to Fitness. Transaction Publishers, New Brunswick (2002)
9. Ariely, D.: Predictably Irrational: The Hidden Forces that Shape our Decisions. Harper Perennial, New York (2007)
10. Ariely, D.: The Upside of Irrationality: The Unexpected Benefits of Defying Logic at Work and at Home. Harper Collins, New York (2010)
11. Christensen, C.: The Innovator's Dilemma. Harper Collins, New York (1997)
12. Simon, H.: The Sciences of the Artificial, 3rd edn. MIT Press, Cambridge (1996)
13. Gregor, S., Jones, D.: The Anatomy of a Design Theory. Journal of the AIS 8(5), Article 2, 312–335 (2007)
14. Williams, K., Chatterjee, S., Rossi, M.: Design of Emerging Digital Services: A Taxonomy. In: Hevner, A., Chatterjee, S. (eds.) Design Science in Information Systems Research, pp. 235–253. Springer, New York (2010)
15. Davis, M.: That's Interesting! Towards a Phenomenology of Sociology and a Sociology of Phenomenology. Philosophy of the Social Sciences 1, 309–344 (1971)
16. Ziliak, S., McCloskey, D.: The Cult of Statistical Significance. University of Michigan Press, Ann Arbor (2008)
17. Mokyr, J.: The Gifts of Athena: Historical Origins of the Knowledge Economy. Princeton Univ. Press, Princeton (2002)

Interface Design Elements for Anti-phishing Systems

Yan Chen, Fatemeh (Mariam) Zahedi, and Ahmed Abbasi

Sheldon B Lubar School of Business
University of Wisconsin–Milwaukee
3202 N. Maryland Ave. Milwaukee, WI 53201, USA
{yanchen,zahedi,abbasi}@uwm.edu

Abstract. Anti-phishing systems are developed to prevent users from interacting with fraudulent websites. However these tools are ineffective since users often disregard their warnings. We present a design science-based assessment of interface design elements for such systems. An extensive taxonomy of important design elements is constructed. A survey is used to evaluate the perceived saliency of various elements encompassed in the taxonomy. The results suggest preferred design elements are in line with efficient information processing of human vision, and indicate that existing tools often fail to consider users' preferences regarding warning design alternatives. The results of users' preference also show the presence of a subset of design elements that could potentially be customized for the population of our sample and others that could be personalized. These findings are being applied in an NSF-supported project, in which we evaluate the impact of customized and personalized warnings on user performance.

Keywords: Anti-Phishing Systems, Interface Design, Warnings, Taxonomy.

1 Introduction

Fake websites are fictional, misrepresentative sites posing as legitimate providers of information, goods, and/or services [1]. Categories of fake websites include concocted sites, spoof sites, and web spam [2]. While web spam sites attempt to deceive search engines to boost their rankings [10], concocted and spoof websites target Internet users. Hence, these latter two categories of fake websites are often collectively referred to as phishing websites, and security tools designed to combat them are commonly called anti-phishing tools [7], [38], [40].

Phishing websites defraud millions of Internet users on an annual basis, generating billions of dollars in fraudulent revenue [7]. Consequently, many security tools have been proposed for combating phishing websites [2]. However, there are important usability issues related to anti-phishing tools [17]. Users often disregard system recommendations, even when they are correct [38] and overlook security warnings because they fail to draw and keep users' attention [7]. Detection methods have little impact unless they are coupled with interfaces most suitable to attract users' attention, and allow them to process the warning with the least amount of time and cognitive efforts.

H. Jain, A.P. Sinha, and P. Vitharana (Eds.): DESRIST 2011, LNCS 6629, pp. 253–265, 2011.

Following the design-science paradigm, in this study we undertake a user-centric examination of critical interface design elements pertaining to anti-phishing systems. Design science provides concrete prescriptions for the development of IT artifacts, including constructs, models, methods, and instantiations [12], [18]. We develop a taxonomy of interface design elements for anti-phishing systems (i.e., a model). When creating IT artifacts in the absence of sufficient design guidelines, many studies have emphasized the need for design theories or existing literature to help govern the development process [19], [32]. Accordingly, we leverage existing studies related to computer warning systems to help inform the development of the taxonomy [23]. A user survey with 412 participants is used to assess the perceived importance of various choices of design elements included in the taxonomy. The expressed preferences are in line with efficient human visual information processing, and have the potential for use in customized and personalized security warning designs.

2 Related Work

While conducting a user study on existing warning tools, Wu et al. ([38], p. 601) observed that users "disregarded or explained away the toolbars' warnings if the content of the web pages looked legitimate." They observed spoof rates as high as 33% to 45% and found that 60% of the subjects used rationalizations to justify discounting warning signs generated by the tool. In another study that used 20 legitimate and phishing websites, 23% of subjects did not look at the web browser or security toolbar cues, resulting in 40% spoof rates [7]. These findings suggest that existing interfaces of detection systems are inadequate in warning users of potential threats.

Obermayer and Nugent in [23] summarized a list of pitfalls of current warning or alarm designs in computer systems. They argued that "many contemporary alerting and warning systems fall short of their goal of providing needed information to the operator in a manner that can be integrated into ongoing tasks; rather, they are often annoying, don't inform, and create havoc with tasks in process" (p. 15). Previous studies also point out that simply applying human-computer interface (HCI) design guidelines to warning interfaces can cause warning delivery failures at various cognitive stages of the human information processing process [21].

Depending on the research perspective, warnings in information systems are categorized differently as *error messages*, *exception messages* [3], *interruptions* [21], [23], and *notifications* [20]. We lack a comprehensive view of human-computer interactions of warnings in relation to various design elements and attributes in the interface. Due to the lack of existing guidelines regarding anti-phishing tools, we developed a taxonomy of interface design elements that provides a foundation for the ontology of design elements for security warning interfaces. The taxonomy is based on an intensive review of the warning delivery literature. A survey approach is used to identify the key features of the security warning interface, and to investigate the potential for customizing and personalizing the security warning interfaces.

3 Warning Design Elements Taxonomy for Anti-phishing Tools

Warnings can be delivered through two modes: visual and/or auditory [35]. Figure 1 and 2 show the proposed taxonomies of visual and auditory design elements, respectively. The table summarizing the supporting literature and related references

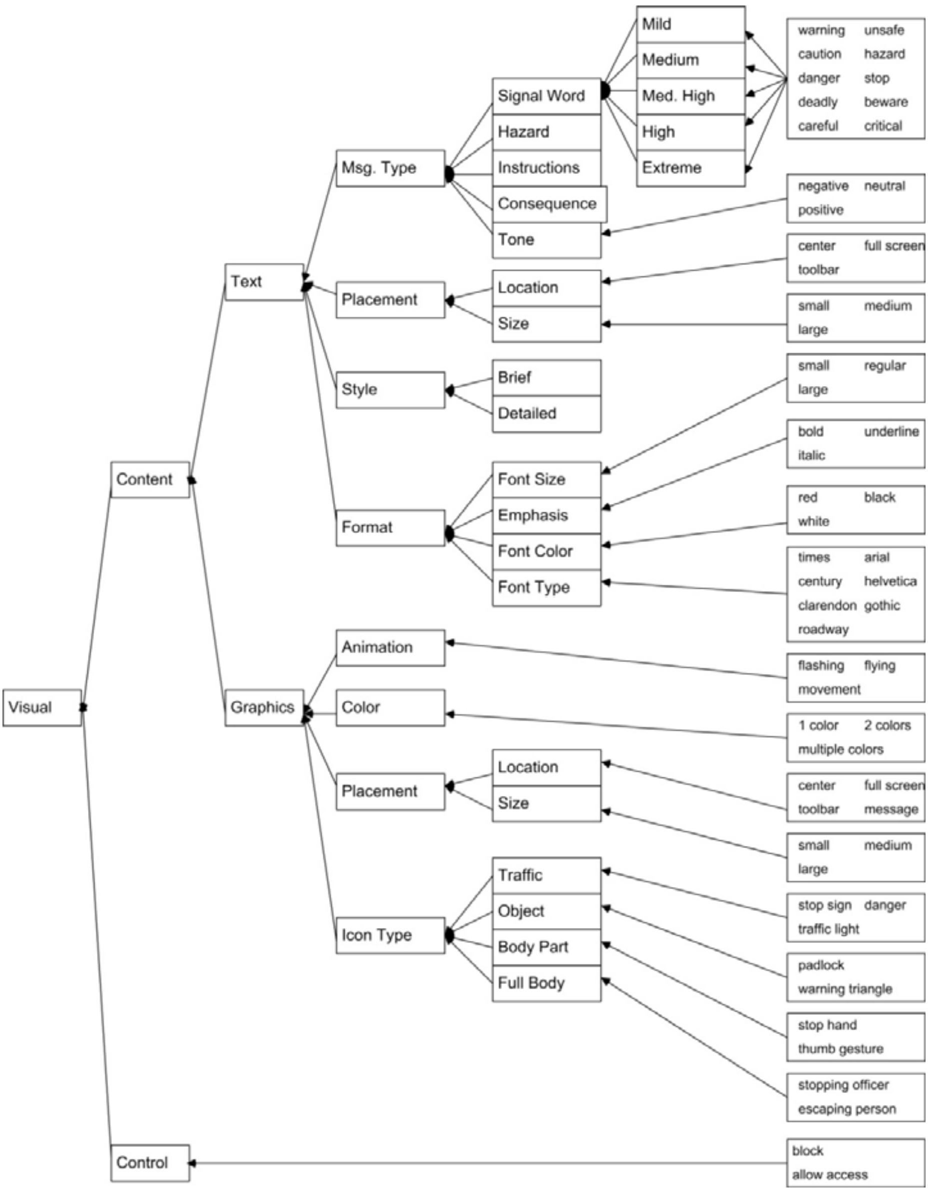


Fig. 1. Taxonomy of Visual Design Elements

were removed due to the space limitation. The visual taxonomy is considerably more extensive since detection tools primarily convey information via visual warning mechanisms. In Figure 1, the boxes on the very right contain example choices for the various visual design elements. For visual warnings (Figure 1), their effectiveness is directly influenced by their *content* in terms of *text* and *graphics* [22], [35]. An effective *warning text* consists of four types of message information: a *signal word*, description of the *hazard*, potential negative *consequences*, and *instructions* on how to avoid the hazard [35], [36]. Additionally, text *format* (e.g., font type, font size, etc.) and *placement* (e.g., location and size) attributes help grab attention and are therefore important design factors for effective warnings [9], [26]. Warning text can also vary in terms of style; while people generally prefer *brief* messages [6], more detailed messages can augment the perceived severity of the threat [28].

Important design considerations for warning graphics include the *color*, *placement*, *icon types*, and the presence/absence of *animation*. Studies have found people prefer warning *icons* that are simple, easy to understand, and that look familiar [11]. Similarly, the *color* and *placement* of graphics can impact the salience of the warning as well as the perceived severity of the threat [23].

The level of *control* given to system operators regarding the warning is a matter of design wisdom [21]. Some have argued that whenever possible, users should be excluded from the security-critical decision loop [4]. For anti-phishing tools, the question becomes whether access to websites deemed illicit should be blocked altogether, or whether users should maintain the option to ignore the system’s warnings.

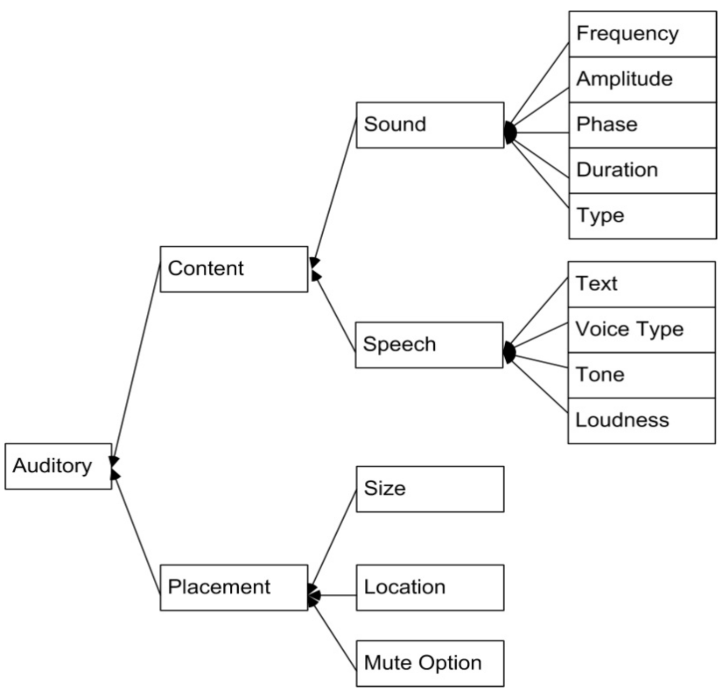


Fig. 2. Taxonomy of Auditory Design Elements

Figure 2 depicts the auditory design element taxonomy. Auditory content elements such as *speech* and *sound* have been found to be important in various warning delivery situations [35]. Both visual and auditory warning mechanisms have their advantages and disadvantages, depending on the application domain [26], [35]. In the following section, we evaluate user preferences regarding the design elements encompassed in our visual design taxonomies. We focus on the visual modality since the visional sense is the most dominant communication channel of human-computer interaction. In addition, people often turn off auditory warnings when they feel the warnings are annoying or distracting [8], [23]. However, the use of auditory warnings could be beneficial in highly critical and urgent situations when visual modality is heavily loaded, or not available [23], [25]. Therefore, we collected data for general preferences for auditory delivery model.

4 Methodology and Data Collection

The next step was to examine users' preferences for design elements within the hierarchy of design-elements taxonomy. The objectives were two-fold: (i) to gain insight about users' preference and importance ratings of design elements, and (ii) to identify design element options that were strongly preferred over others, indicating the presence of a standard for customization of design elements for the population of our sample. The lack of an emerging winner for the design element choices would indicate the potential for personalization of design elements.

Following [30], a survey instrument was developed to collect data for users' preferences for design element categories and their choices. The focus was on the details of the visual branch of the taxonomy (vis-à-vis the auditory branch) since this category is a more preferred method of delivery [8], [23]. Size was operationalized for the font size (10pt, 10-12pt, and >12pt) and the message box (the percentage of the screen covered: 10%, 40%, and 80%). Data was collected for the third level of the taxonomy hierarchy and the elements below them since these are more concrete from a user's perspective.

Table 1. Participant Demographics, N=412

Profile Variables	Mean	Std. Deviation	
Age*	1.12	.42	
Education**	3.02	.64	
Time spent on the Internet***	3.82	1.06	
Gender	Male= 55.4%	Female= 44.6%	
Browser I frequently use has fake website detection function	Yes= 32.0%	No=10.7%	Don't know=57.3%

*Age scales: 1=18-24 (90.5%), 2=25-34 (6.6%), 3=35-44 (1.5%), 4=45-54 (0.7%), 5=55-64 (0), 6=Over 65 (0) years old.

**Education scales: 1= Some school, no degree (0.5%), 2= High school graduate (11.4%), 3= Some college, no degree/college students (78.2%), 4= Professional degree/2-year associate degree (5.1%), 5=Bachelor's degree (3.9%), 6= Master's degree (0.2%), 7=Doctoral degree (0.2%).

***Time scales: 1=Less than 4 hours per month (0.2%), 2=1 hour or less per day (4.4%), 3=1-3 hours per day (39.3%), 4=3-5 hours per day (34.5%), 5=5-7 hours per day (15%), 6=7-9 hours per day (3.6%), 7=More than 9 hours per day (2.9%)

Participants were asked to rate the design elements. The measurement scale was a semantic differential scale from 1-not important at all to 10-very important for sure. The data was collected through a web-based survey. Of the 850 students in a large Midwestern university invited, 432 students took the survey (a 51% response rate). To ensure the validity of responses, the data was cleansed by deleting the records lacking responses about design elements and those that had taken under 5 minutes to complete the survey. The 5-minute criterion was established based on our pre-tests showing that a minimum of five minutes was needed to read and answer questionnaire items. This resulted in 412 usable responses. Table 1 reports participants' demographics.

5 Analysis and Results

To examine the importance of elements, following [30], the mean, standard deviation and standard error were computed for each design element, as reported in Table 2. Pairwise t-tests were performed to examine whether within each category of design elements, there is an element that is significantly preferred to others within each category, indicating the presence of a potential standard for the population of respondents. Figure 3 reports the importance ratings of design elements for the visual design elements.

The results showed that the participants had a clear preference for visual (text and graphics) over auditory (sound and speech) for the warning media. They rated text as the most critical among the four design elements, text, graphics, sound, and speech. In the visual branch, text was the preferred mode. Message type and location of the message were more important for users than format or style of message. For the location of the message, users indicated a strong preference for locating both the text and graphs in the center of the screen. Users preferred a blocking warning system that requires a user action to unblock. Thus, the emerging standard for customizing the interface for our sample population is an interface that has a medium size message box in the center of the screen, uses a short message with a negative tone, has red as the dominant color, and blocks users from going to potentially harmful websites.

While the general structure of the warning interface indicates the presence of a standard for customization of design elements for a user population, the results for other design elements show the potential for personalization. For example, the close mean values and high standard errors for arousal strength of the warning message indicate that there might be the opportunity to personalize the arousal strength of the message based on users' general typology (such as experts vs. novices) or personalize based on users' demography or personality.

Similarly, message format, font size, and emphasis may have room for personalization. Similar opportunity may exist for personalizing the choice for icon types and animation. Our results are in line with the human information processing capacities of users. The security-warning interface should be effective in communicating the warning message by minimizing human's information processing time and cognition load. Our findings indicate that the elicited preferences are indeed in the direction of optimum information processing—the neurobiology of human visual perception and the psychology of dealing with a challenge or threat.

Table 2. Results (N=412)

Variables	Mean	Std. Error	Std. Dev.	Sig. of Diff. P-value[a]	Variables	Mean	Std. Error	Std. Dev.	Sig. of Diff. P-value[a]
Main Components					Emphasis				
Text	9.48	.10	2.05		*Bold*	8.67	.15	3.16	
Graph	8.25	.13	2.53	.000***	Underlining	7.25	.17	3.47	.000***
Sound	6.73	.16	3.14	.000***	Italic	5.41	.17	3.37	.000***
Speech	6.10	.16	3.19	.000***	Font Color				
Control					*Red*	8.07	.17	3.43	
Control	8.50	.11	2.19		Black	7.49	.17	3.38	.008**
Block	8.04	.14	2.84		White	4.23	.15	3.11	.000***
Unblock	6.37	.15	3.03	.000***	Font Type				
Text					*Times New Roman*	6.55	.20	3.98	
Message Type	8.85	.10	1.94		Arial	5.78	.18	3.58	.000***
Placement	8.68	.11	2.16	.087^{ns}	Helvetica	5.31	.17	3.39	.000***
Format	8.16	.13	2.57	.000***	Century Schoolbook	4.74	.16	3.32	.000***
Style	7.87	.11	2.28	.000***	Clarendon	4.73	.16	3.33	.000***
Message Type					Rawlinson Roadway	4.61	.16	3.28	.000***
Instruction	8.94	.11	2.22		Highway Gothic	4.55	.16	3.25	.000***
Hazard	8.85	.11	2.13	.464^{ns}	Style				
Signal Words	8.78	.10	2.02	.237^{ns}	*Short in Outlines*	9.10	.11	2.13	
Consequences	8.68	.12	2.32	.002**	Long in Details	5.24	.14	2.81	.000***
Tone	8.13	.11	2.18	.000***	Graphics				
Arousal Strength					*Placement*	8.13	.11	2.27	
High	8.58	.13	2.59		Icon Type	7.96	.12	2.47	.115^{ns}
Medium High	8.18	.12	2.45	.007**	Color	7.49	.13	2.66	.000***
Extreme	7.77	.17	3.38	.000***	Animation	6.97	.15	2.94	.000***
Mild	7.19	.13	2.59	.000***	Placement				
Medium	6.10	.15	2.95	.000***	*Location*	8.17	.11	2.30	
Signal Words					Size	7.90	.12	2.34	.000***
Warning	9.12	.11	2.31		Location				
Caution	8.66	.12	2.44	.000***	*Center of the Screen*	8.12	.135	2.72	
Danger	8.33	.14	2.89	.000***	Message Box	7.16	.142	2.88	.000***

Table 2. (*Continued*)

Stop	8.32	.14	2.77	.000***	Full Screen	6.66	.17	3.37	.000***
Unsafe	7.98	.14	2.78	.000***	Toolbar	5.68	.15	3.07	.000***
Beware	7.83	.14	2.90	.000***	Size				
Hazard	7.78	.15	2.98	.000***	*Medium*	7.53	.13	2.59	
Critical	7.63	.15	3.08	.000***	Large	6.87	.16	3.18	.000***
Dead	6.69	.19	3.78	.000***	Small	5.81	.15	3.05	.000***
Careful	6.41	.15	3.09	.000***	Icon Type				
Tone					*Traffic Sign*	7.40	.14	2.80	
Negative	7.66	.15	2.95		Object/Symbol	7.19	.13	2.69	.164ns
Neutral	7.16	.15	3.00	.012*	Body Part	6.99	.14	2.87	.001*
Positive	6.79	.14	2.90	.000***	Full Body	5.16	.16	3.26	.000***
Placement					Traffic Sign				
Location	8.83	.11	2.13		*Stop Sign*	8.97	.12	2.51	
size	8.49	.11	2.14	.000***	Danger Zone	6.57	.17	3.39	.000***
Location					Traffic Lights	6.25	.16	3.28	.000***
Center of the Screen	9.08	.12	2.34		Object/Symbol				
Full Screen	7.33	.16	3.28	.000***	*Danger Triangle*	7.92	.14	2.85	
Toolbar	5.86	.15	3.11	.000***	Padlock	6.14	.16	3.22	.000***
Size					Human Image-Partial				
Medium	8.11	.12	2.46		*Thumbs Gesture*	7.95	.15	2.99	
Large	7.45	.15	3.01	.000***	Hand-Stop	6.30	.16	3.15	.000***
Small	5.50	.16	3.13	.000***	Human Image-Full				
Format					*Stopping Officer*	5.82	.17	3.46	
Font Size	8.57	.14	2.78		Escaping Person	4.78	.16	3.26	.000***
Emphasis	8.39	.14	2.88	.085ns	Color				
Font Color	8.12	.15	2.30	.000***	*2 Colors*	7.12	.14	2.89	
Font Type	6.63	.17	3.47	.000***	Multiple Colors	6.48	.17	3.34	.000***
Font Size					1 Color	5.92	.16	3.25	.000***
Large	8.59	.14	2.83		Animation				
Regular	6.67	.15	2.30	.000***	*Flashing*	7.58	.15	3.09	
Small	3.77	.14	2.88	.000***	Flying	5.99	.16	3.22	.000***
					Movement	5.96	.16	3.18	.000***

[a] The mean difference of each element within a group is tested for its significant difference from the element with the highest mean in that group. In each category, the element with the highest mean value is shown in italics.

*p<0.5, **p<0.01, ***p<0.001, ns not significant.

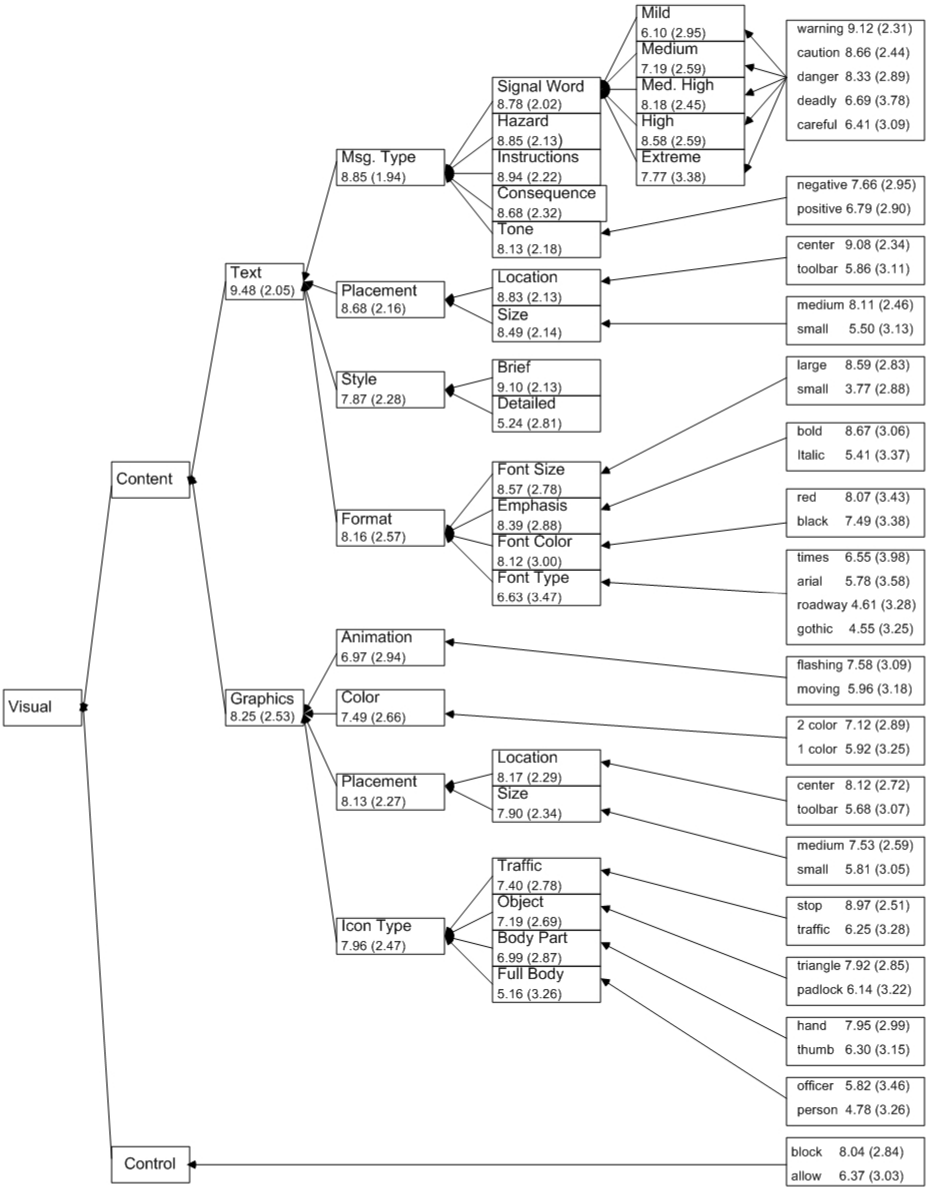


Fig. 3. Survey Results for Visual Design Elements

(1) The preference for a message box in the center increases the speed of perception in human eye. The human retina contains rod and cone opsin protein molecules as photoreceptors [24], [33], [37]. Cone photoreceptors operate in light, and process color vision, whereas rod cells operate in dim light and dark, and do not allow color perception. The center of the retina, called the fovea centralis (or fovea for

short), is one millimeter wide, has the highest concentration of cone photoreceptors, has the fastest processing paths, has 50% more optic nerves, and allows for the most detailed perception. A 20 degree distance from the fovea reduces vision acuity by 80%. Since the default gaze of users is on the center, a centered warning message box provides the most efficient path for processing the warning information.

(2) The preference for a medium size message box can also be explained by the neurobiology of the human visual system. Peripheral vision (parafovea and perifovea cells at 1.25 and 2.75mm distance from the fovea) contains rod cells, which are far less precise and far slower in processing visual stimuli [13]. A large security-warning window requires users to shift their gaze in order to traverse the message box in search of important information. This increases the information processing time. A smaller size message box would rely on fovea cells without the need for shifting the gaze. However, a very small size could increase the cognitive load for recognition of the components of the message. Hence, the results indicate the possibility of an optimal size for the message box for maximizing the speed of the message processing. This is also in line with the process of "foveated imaging" used in compressing digital images and reducing the load [34]. In foveated imaging, there are fixed points, which contain more pixels and details, whereas the peripherals of the fixed points have reduced pixels and are blurred. The centered message box is similar to the fixed point of a foveated image. The peripheral parts should contain reduced information load since in human peripheral vision, rods have reduced information processing capability and lack adequate capacity for the perception of details. However, they are more efficient in perceiving movement. It is possible that animation could be located on the tool bar or left or right end of the screen since the peripheral vision would process flashing icons more quickly. This is an area that needs further investigation.

(3) Preference for the color red is not a new finding and is also in line with the neurobiology of human vision. In human computer interface design, it is already acknowledged that humans have limitations in their color perception [15]. Color perception in the human retina involves cone cells and three independent channels: blue, green and red, which operate based on the Trichromatic Color Theory developed by Von Helmholtz in 18th century [16]. The brain perceives color by addition or subtraction of signals from these three channels. The red channel cones respond to the long spectra of light (L), whereas green is a moderate (M) and blue is a low (S) channel type. It is found that the L and M channels are a more recent evolution in primates [14]. It is also shown that there are fewer S cones and they are randomly placed, whereas M cones and particularly L cones have a higher frequency in human eyes [27], making the red channel a faster processing channel. In different cultures, red has been the sign of danger, risk, celebration, and happiness—all indicating attention arousal [39]. Therefore, preference for red is in line with the need for quick attention arousal for efficient processing of information. However, intensity and color contrast are other parameters for efficient color perception that have not been investigated for security-warning interfaces, an area that requires further investigation.

(4) The results indicate that users preferred moderate high-intensity signal words. Such words produce high arousal, which captures immediate attention. According to Csikszentmihalyi [5], in response to a challenge, there is a spectrum from (i) anxiety to (ii) arousal to (iii) flow, depending on individuals' skills in dealing with the

challenge. Considering that the respondents were at neither very low nor very high skill extremes in dealing with Internet security, they preferred words that produced attention arousal signal words, but not extreme words that could produce anxiety. We expect that this preference may vary with users' skill levels. For example, highly skilled users or those who do not feel threatened by security risks may prefer lower intensity signal words. This is another area that needs further research.

Our results are in contrast with some of the existing tools, which put the warning message on the toolbar, do not require user action to unblock, and use small size messages. For instance, many existing tools use warnings that are displayed on a toolbar or via a message box, despite users' preferences for alternate methods [38], [40]. Moreover, most existing tools do not allow any customization of visual design elements. In a study involving legitimate and spoof banking websites, subjects exhibited dramatically reduced spoof rates when using a customizable toolbar: one where users were allowed to select their own warning icons/images [11]. It has been observed that customization and personalization increase information congruity, enhance attention, and facilitate the decision-making process [31]. Congruent information leads to a more a favorable attitude [29]. Hence, we expect that using our results to customize and personalize security interface may promote favorable attitude towards security tools and enhance their effective use. This is an area of further inquiry.

6 Current and Future Work

In this work, we developed a taxonomy for design elements for the interface of anti-phishing systems using an extensive review of the existing literature. In order to determine the relative importance of design elements and the potential existence of a set of highly preferred elements for a security warning interface, we undertook collecting data through a survey research method. The results revealed that many users have distinct preferences regarding the design elements used in the general structure of the interface, indicating the potential presence of a standard for customizing interface design for the population of our sample. Certain design elements on the lower level of the taxonomy showed potential for personalization of the interface. We found preferences for a number of design elements to be at odds with the design of existing security warning tools. On the other hand, our findings were in line with the neurology of human visual information processing, indicating that the warning interface could be designed to maximize its processing speed and minimizing cognitive load, hence making the interface more effective and users more responsive to them. Furthermore, our findings suggest that there is a need for preferential congruence, which may be facilitated through customized/personalized interfaces. This study was carried out within the context of a large funded project in studying the design of anti-phishing systems. The taxonomy and findings are being used to explore whether anti-phishing systems equipped with customized and/or personalized interfaces enhance usability and reduce fraud rate as compared with existing alternatives.

Acknowledgements. This material is based work supported by the National Science Foundation under Grant No. CNS-1049497.

References

1. Abbasi, A., Chen, H.: A Comparison of Tools for Detecting Fake Websites. IEEE Computer 42(10), 78–86 (2009)
2. Abbasi, A., Zhang, Z., Zimbra, D., Chen, H., Nunamaker Jr., J.F.: Detecting Fake Websites: The Contribution of Statistical Learning Theory. MIS Quarterly 34(3), 435–461 (2010)
3. Amer, T.S., Maris, J.B.: Signal Words and Signal Icons in Application Control and Information Technology Exception Messages—Hazard Matching and Habituation Effects. Journal of Information Systems 21(2), 1–26 (2007)
4. Cranor, L.F.: A framework for Reasoning about the Human in the Loop. In: Proc. of 1st Con. on Usability, Psychology, and Security, pp. 1–15. USENIX Association, Berkeley (2008)
5. Csikszentmihalyi, M.: Finding Flow: The Psychology of Engagement in Everyday Life. Basic Book, New York (1997)
6. Desaulniers, D.R.: Layout, Organization, and the Effectiveness of Consumer Product Warnings. In: Proc. Human Factors Society 31st Annual Meeting, Santa Monica, CA, pp. 50–60 (1987)
7. Dhamija, R., Tygar, J.D., Hearst, M.: Why Phishing Works. In: Proceedings of the ACM Conference on Computer Human Interaction, Montreal, Quebec, pp. 581–590 (2006)
8. Edworthy, J.: The Design and Implementation of Non-Verbal Auditory Warnings. Applied Ergonomics 25(4), 202–210 (1994)
9. Edworthy, J.: Warnings and Hazards: An Integrative Approach to Warnings Research. International Journal of Cognitive Ergonomics 2(1/2), 3–18 (1998)
10. Gyongyi, Z., Garcia-Molina, H.: Spam: It's not Just for Inboxes Anymore. IEEE Computer 389(10), 28–34 (2005)
11. Herzberg, A., Jbara, A.: Security and Identification Indicators for Browsers Against Spoofing and Phishing Attacks. ACM Transactions on Internet Technology 8(4), article #16 (2008)
12. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28(1), 75–105 (2004)
13. Iwasaki, M., Inomara, H.: Relation between Superficial Capillaries and Foveal Structures in the Human Retina. Investigative Ophthalmology & Visual Science 27, 1698–1705 (1986)
14. Jacobs, S., Nathans, J.: The Evolution of Primate Color Vision. Scientific America, 32–39 (April 2009)
15. Johnson, J.: Designing with the Mind in Mind. Morgan Kaufmann Publishers, Burlington (2010)
16. Kahl, R. (ed.): Selected Writings of Herman Von Hemlholtz. Wesleyan University Press, Middleton (1971)
17. Li, L., Helenius, M.: Usability Evaluation of Anti-Phishing Toolbars. Journal in Computer Virology 3(2), 163–184 (2007)
18. March, S.T., Smith, G.: Design and Natural Science Research on Information Technology. Decision Support Systems 15(4), 251–266 (1995)
19. Markus, M.L., Majchrzak, A., Gasser, L.: A Design Theory for Systems that Support Emergent Knowledge Processes. MIS Quarterly 26(3), 179–212 (2002)
20. McCrickard, D.S., Chewar, C.M., Somervell, J.P., Ndiwalana, A.: A Model for Notification Systems Evaluation—Assessing User Goals for Multitasking Activity. ACM Transactions on CHI 10(4), 312–338 (2003)

21. McFarlane, D.C., Latorella, K.A.: The Scope and Importance of Human Interruption in Human–Computer Interaction Design. Human-Computer Interaction 17(1), 1–61 (2002)
22. Monaghan, S., Blaszczynski, A.: Electronic Gaming Machine Warning Messages: Information versus Self-Evaluation. The Journal of Psychology 144(1), 83–96 (2010)
23. Obermayer, R.W., Nugent, W.A.: Human-Computer Interaction for Alert Warning and Attention Allocation Systems of the Multi-Modal Watchstation. In: Proceedings of the International Society for Optical Engineering (SPIE), Bellingham, WA, pp. 14–22 (2000)
24. Osterberg, G.: Topography of the Layer of Rods and Cones in the Human Retina. Acta Ophthalmoogica Supplement 13(6), 1–102 (1935)
25. Patterson, R.D., Mayfield, T.F.: Auditory Warning Sounds in the Work Environment [and Discussion]. Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences 327(1241), 485–492 (1990)
26. Rogers, W.A., Lamson, N., Rousseau, G.K.: Warning Research: An Integrative Perspective. Human Factors: J. Human Factors and Ergonomics Society 42(1), 102–139 (2000)
27. Roorda, A., Williams, D.R.: The Arrangement of the three Cone Classes in the Living Human Eye. Nature 397, 520–522 (1999)
28. Silver, N.C., Leonard, D.C., Ponsi, K.A., Wogalter, M.S.: Warnings and Purchase Intention for Pre-Control Products. Forensic Reports 4, 17–33 (1991)
29. Smith, S.M., Fabrigar, L.R., Powell, D.M., Estrada, M.-J.: The Role of Information-Processing Capacity and Goals in Attitude-Congruent Selective Exposure Effects. Pers. Soc. Psychol. Bull. 33(7), 948–960 (2007)
30. Song, J., Zahedi, F.M.: A Theoretical Approach to Web Design in E-Commerce: A Belief Reinforcement Model. Management Science 51(8), 1219–1235 (2005)
31. Tam, K.Y., Ho, S.Y.: Understanding the Impact of Web Personalization on User Information Processing and Decision Outcomes. MIS Quarterly 30(4), 865–890 (2006)
32. Walls, J.G., Widmeyer, G.R., El Sawy, O.A.: Building an Information System Design Theory for Vigilant EIS. Information Systems Research 3(1), 36–59 (1992)
33. Wandell, B.A.: Foundations of Vision. Sinauer Associates, Sunderland (1995)
34. Wang, Z., Lu, L., Bovik, A.C.: Foveation Scalable Video Coding with Automatic Fixation Selection. IEEE Transactions on Image Processing 12(2), 243–254 (2003)
35. Wogalter, M.S. (ed.): Handbook of Warnings. Lawrence Erlbaum Associates, Mahwah (2006)
36. Wogalter, M.S., Conzola, V.C., Smith-Jackson, T.L.: Research-Based Guidelines for Warning Design and Evaluation. Applied Ergonomics 33(3), 219–230 (2002)
37. Wyszecki, G., Stiles, W.S.: Color Science: Concepts and Methods, Quantitative Data and Formulae, New York, NY. Wiley Series in Pure and Applied Optics (1982)
38. Wu, M., Miller, R.C., Garfunkel, S.L.: Do Security Toolbars Actually Prevent Phishing Attacks? In: Proceedings of the Conference on Human Factors in Computing Systems, Montreal, Canada, pp. 601–610 (2006)
39. Zahedi, F.M., Bansal, G.: Cultural Signifiers of Web Images. Journal of Management Information Systems (2011) (forthcoming)
40. Zhang, Y., Egelman, S., Cranor, L., Hong, J.: Phinding Phish: Evaluating Anti-phishing Tools. In: Proc. 14th Network and Distributed System Security Symposium, San Diego, CA (2007)

Experimental Evaluation of Peer Endorsement System Artifacts Using Best-of-Breed Ideals – Effects of Online Decision Confidence on Post-choice Regret

Mitchell Church and Lakshmi Iyer

University of North Carolina at Greensboro – ISOM Department,
1400 Spring Garden St
Greensboro, North Carolina 27412
{emchurch,Lsiyer}@uncg.edu

Abstract. Peer endorsement systems (PES), systems for the collection and presentation of online product reviews represent a new and increasingly common sight on many e-Commerce websites. The number of reviews and scope of data presentation made possible by PESs pose demands on the information processing abilities of a typical customer. This study posits that differences in PES design based on quality and presentation of online customer reviews can impact user perceptions about whether they are getting "the whole story" about a product. Drawing on economic regret theory, we develop a theoretical model to test the impact of PES design on consumer's decision confidence and post-choice regret. We conduct an expost experimental analysis of two competing PES interfaces and show that key differences in PES design have a surprising and relevant impact on the way in which data from that interface is perceived and used by customers in the online decision-making process.

Keywords: Peer endorsement system, design science, regret, decision confidence,artifact design.

1 Introduction

Electronic commerce websites today compete for consumers by presenting detailed product information, customer review information, pictures, blogs, and similar product recommendations. Improvements in hardware and network technology make it possible today to display far more information than a human being can absorb and make use of. This glut of information creates information overload as information vital to good decision making gets lost in the surrounding noise [1,2]. Despite the well understood impact of information overload, which has been shown to lead to poor decision making [3] and general product dissatisfaction [1], websites continue to emphasize the display of more and more information in their design [4].

Peer endorsement systems (PES), systems for the collection and presentation of online product reviews [5] represent a new and increasingly common sight on many e-Commerce websites. These systems, which are conceptualized as distribution agents

H. Jain, A.P. Sinha, and P. Vitharana (Eds.): DESRIST 2011, LNCS 6629, pp. 266–275, 2011.

for electronic word of mouth (WOM) advertising [6,7] have been shown to play an important role in customer opinion formation, and thereby directly influence product sales [8,5]. However, the sheer number of reviews, and the scope of data presentation made possible by a PES, poses certain demands on the customer. It is not at all uncommon to see products on sites like Amazon.com with thousands of customer reviews. Since no customer can realistically analyze and interpret such a large amount of information, methods are needed to distinguish the most helpful review information available [9].In an attempt to understand a website's impact on the online decision-making process, we draw upon theory related to economic regret. In the economics literature, regret is conceived as a negative emotion borne of a choice made as it relates to another better option that may have been missed by the consumer [10] due to information overload. Regret is appropriate for this study, as it is fundamentally tied to the quality and availability of information used in arriving at a decision decision[11,12]. As information quality increases, the likelihood that additional important information has not been lost in the shuffle is reduced. Research has shown that the anxiety that arises over the possible existence of this kind of important information is what ultimately drives feelings of regret [13,14]. We posit that differences in PES design can fundamentally impact information quality, and user perceptions about whether they are getting "the whole story" about a particular product.

Regret has garnered little attention from the IS research community [15]. This may be due to the difficulty in obtaining accurate measures of regret, since feelings of regret, if they come at all, lag behind any actual decision making process. In this study, we attempt to get around this difficulty by capturing a measure of perceived likelihood of post-choice regret. Thus, it is ultimately the PES's ability to influence the potential for regret that is of interest. Low levels of product and decision confidence at the time of decision have been shown in past studies to be good estimators of future post-choice regret [14]. Hence, we examine the effect decision and product confidence measures collected at the time of the decision have on post-choice regret. Thus, the following two general objectives drive our research:

- *What is the impact of PES design on feelings of consumer confidence in the e-Commerce decision making process?*
- *What are the ideal characteristics of a theoretical "next-generation" PES designed specifically for this purpose?*

Baskerville Pries-Heje, and Ramesh, [16] argue that rigorous evaluation of new and existing artifacts is important to ensure that these innovations meet their expressed goals in as effective a manner as possible. To guide the development of new PES design artifacts, it is important to examine existing PES and/or protoypes to determine what should be the characteristics of new PES design [17]. Experimental ex-post evaluation allows design science researchers to make grounded assumptions about the direction that artifact design should take [18].

In this study, we examine two best of breed PES systems (Amazon.com and Google Products) to identify the features of each and areas of concern that may prevent each of these technologies from accomplishing their expressed goal of an improved customer information search experience as measured by product/decision choice confidence and post-choice regret.

2 Theoretical Foundations

2.1 Peer Endorsement Systems, Product Choice Confidence, and Decision Confidence

Mudambiand Schuff [5] define online customer reviews as "peer-generated product evaluations posted on company or third party websites". Together with automated recommendation systems and other provided product information, they serve the purpose of providing customers with as clear a picture as possible of the features of a product. Unlike these other systems, however, they serve to aggregate and present the product in the context of the user community itself.

Mere presence of a PES has been shown to improve customer perceptions of a website's usefulness [19]. PES can be found today on nearly all major e-Commerce websites. For this reason, the presence of a PES today may not provide the same distinction as it once did. Thus, other factors related to PES, namely the informational value of the review content it provides, become paramount [5]. Mudambi and Schuff[5] conceptualized the informational value of an online review in terms of its helpfulness to Amazon.com customers. Review helpfulness can be thought of as a review's ability to aid the information search phase of the customer decision-making process. Presentation format and interface design thus play a part in the overall helpfulness of the PES implementation.

Product Choice Confidence refers to a customer's perception that they have complete and accurate information with which to make a product selection [14]. Put another way, customers with high product choice confidence feel that there is not information regarding product quality or appropriateness that they failed to, or were unable to consider in their information search for that particular product. Product choice confidence has significant implications for e-Commerce retailers [12,20], and the link between customer expectations and actual product experience is well understood. In the IS literature, this is known as expectation disconfirmation. Failing to meet expectations has been shown to lead to a decline in online trust, perceptions of the system, and intention to transact additional business with the retailer [20].

Decision Confidence is concerned with a customer's perceptions regarding the logic or wisdom used in making a decision [13]and depends largely on the quality of the customer's information search and the methods used when arriving at the decision. An e-Commerce customer overlooks a suitable alternative product during the purchase decision-making process. Eventually, the later discovery of this product presents a better fit for a customer need, which causes the customer to doubt the quality and wisdom of their decision. Confidence here is unrelated to the original product chosen. If we assume this product was accurately advertised, and customer expectations wholly matched the eventual usage experience, then we must conclude that consumer confidence in this case is separate and distinct from any specific product. Thus, the confidence a customer has in the decision-making process is itself a unique construct [13,14].

Research on the economic impact of regret dates back to regret theory put forth by Loomes and Sugden [10]. Under regret theory, regret is a negative feeling borne of an action one takes that they subsequently wish they had not [12]. Thus, regret is the realization, post-choice, that a different action might have led to a better outcome [14].

Das and Kerr [14] conceptualized regret as deriving from two main antecedents. Regret based on the quality of a purchased product results from a Post-choice comparison of that product and some previously unknown information that creates valence between the product chosen and its (previously available, now lost) alternative [12].

Clearly, the accepted definition of regret suggests a natural relationship between feelings of regret and the quality of information used in decision-making. Individuals with access to more and better information are aware of more options, and better equipped to evaluateunselected alternatives [11,21]. Regret has been examined in some IS studies that looked at Ebay auction formats [22] and adverse selection of luxury goods [23].

2.2 PES Design Comparison

Often, PES review results can number in the hundreds to thousands. Users are provided with only basic search functionality to look for key terms, but no ability to summarize or look for common trends in reviews is provided. A challenge in current PES research is concerned with how this large body of unstructured data can be summarized in such a way as to generate meaningful and useful information [6,7,24]. Originally, we sought to contribute to this review by developing a design science artifact that would examine this unstructured data and return both relevant keywords and phrases that customers could use for both product evaluation and future information search. During this design process, it was brought to our attention that Google Products (PES2) was currently beta testing a product with many of the same functionalities. At that time, the focus of this research shifted from artifact design to an ex post comparison of these two divergent systems. Hence for this study, we compare the PES interface of Amazon.com (PES1) with that of Google Products (PES2).

The PES1 system provides reviews for nearly every one of Amazon's hundreds of thousands of products. It is not uncommon for a product on Amazon.com to garner over one thousand reviews. Each page in PES1 contains information detailing the total number of reviews, complete with a breakdown of reviews by star rating (1-5 stars) and the cumulative overall star rating given to a product. The star rating is the average rating (again 1-5 stars) across all reviews for that particular product. The system also allows a user to flag a review as helpful or not helpful, with the most helpful reviews being displayed first. Although the system does provide some basic overview information, in general PES1 makes use of a narrative format. Reviews are provided to customers without summarization, and users are able to read as many or as few reviews as they wish.

After examining the functionality provided by PES2, we set up an experiment to compare this system with the PES1 website interface. PES2, which at the time of writing is still in a beta stage, provides aggregate review information taken from a number of retailers, including Amazon.com. The system relies on Google's core competency of information search to present a large amount of summary data in a concise format.

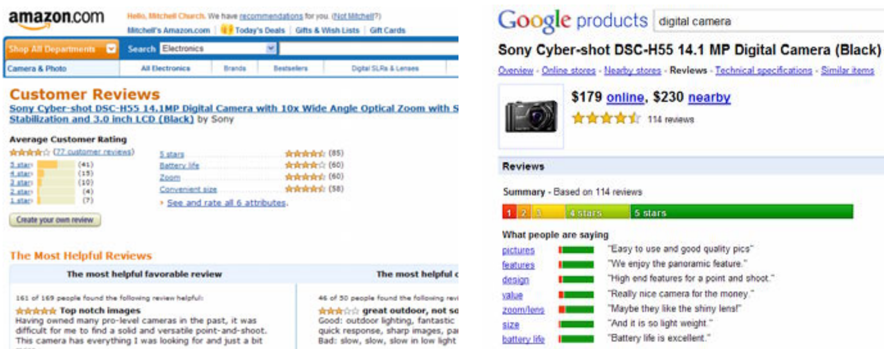


Fig. 1. Example of PES1 and PES2 Results

With PES2, the emphasis is not on reading entire reviews in a narrative form. Rather, users are provided with quotations taken from reviews that are tied to key terms. For example, the total review content for a movie may contain frequent occurrences of the word "romantic". PES2 allows users to view reviews in which this keyword appears.

For this study, we used an ex-post design science methodology [18] to experimentally test the performance of each of these interfaces in a simulated e-Commerce environment. This was done to show the merits and deficiencies of each interface in regards to our variables of interest. In the following section, we discuss the theoretical underpinnings that determined our design decisions.

2.3 Theoretical Model

The theoretical model (Figure 2) shows the proposed relationship between the constructs under analysis. Product choice confidence refers here to user perceptions that the information provided by the PES was adequate to accurately gauge the product considered. Decision confidence refers here to user perceptions that the information provided by the PES could be used to make a reliable and accurate decision. Finally, Post-choice regret refers to user perceptions, post-purchase, that the decision process will prove sound upon further reflection.

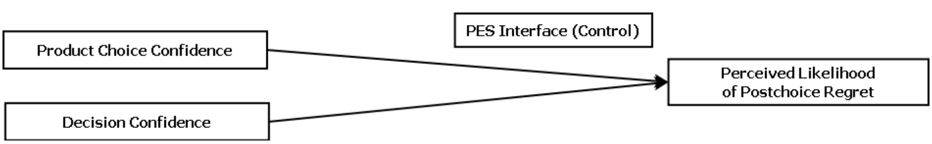


Fig. 2. Theoretical model of post-choice regret in ecommerce

3 Empirical Evaluation

Experimental evaluation provides numerous useful benefits in design science [17]. Empirical experiments test artifact utility, as well as allow for greater generalizability of results. In this study, we simultaneously evaluate the performance of two current PES iterations to identify the strengths of each. This expost design allows for the conceptualization of a future third artifact ultimately containing the best characteristics of the two [18].

We argue here that a PES can either promote or hinder the helpfulness of an online customer review. The interface design itself either brings helpful information to the front of the customer's attention, or hides it in the background among other, less pertinent data. When a PES interface performs its job correctly, it allows the information that most effects Product and Decision confidence to bubble to the top. Of the two interfaces studied, Google Products (PES2) provides a means of filtering relevant information from the larger mass of customer review data. This filtering does not come without a cost. In the process of summarizing large amounts of review data, the narrative context inherent to the individual reviews is lost. Therefore, a major question of interest here is whether this filtering process will actually lead to better information quality, or is there perhaps some tangible value to the consumer in preserving the original narrative dialogue as presented by Amazon's (PES1).

We propose that a PES built to reduce information overload should provide the customer with a higher level of awareness regarding information that would erode confidence in the products they buy, or the purchase decisions they make. Thus, the interface that elicits feelings of both Product and Decision confidence should also lead to lower perceptions of post-choice regret. Based on our proposed theoretical model we have the following hypotheses:

- *H1: Product choice confidence is negatively correlated with total post-choice regret.*
- *H2: Decision choice confidence is negatively correlated with post-choice regret.*
- *H3a: Product choice confidence will differ between PES1 and PES2 users.*
- *H3b: Decision choice confidence will differ between PES1 and PES2 users.*
- *H3c: Perceived Likelihood of Post-choice Regret will differ between PES1 and PES* 2 users.

3.1 Experiment Design and Data Collection

A total of 242 subjects participated in the experiment. The study was conducted in an institution of higher-education. Numerous studies in the area of e-Commerce have justified the use of students as study participants because of as a group they are representative of the typical e-Commerce user [25,26]. Of these participants, all subjects were familiar with buying products online, and peer endorsement systems in general. 77% were female, with 54% making less than $40,000 a year. 56% reported working with or using computers more than four hours a day.

The specific nature of the experiment was as follows: Participants were selected from a pool of undergraduate students, and then randomly assigned into one of two treatment groups. After answering a set of demographic questions, participants were shown a set of three product pages in either the Amazon.com or Google Products PES. Subjects were allowed to peruse the reviews, and were given no instruction into the way in which they should use the system (i.e. aspects of the interface were not emphasized by the researchers). At the end of the allotted time, participants answered a short series of questions designed to measure their feelings of product and decision satisfaction, as well as post-choice regret. In order to avoid confusion, reverse coded questions were avoided wherever possible. For this reason, higher scores on all variables represent positive feelings. Thus high post-choice regret scores are actually associated with lower perceptions of post-choice regret.

3.2 Hypothesis Testing

First, we tested whether there is statistical support for the proposed theoretical relationships between product choice confidence, decision confidence, and post-choice regret. For this hypothesis test we first used an analysis of variance (ANOVA) technique to test the overall suitability of the model. Support was found for this test (F value 194.37; p-value=.0001). We next regressed the product choice confidence and decision confidence on the dependent variable post-choice regret.

H1 was supported (t value 2.77; p-value=0.007). H2 was also supported (t value 9.44; p-value=.0001). Holding decision confidence equal, a one unit increase in product choice confidence was associated with a .24 point increase in post-choice regret. Holding product choice confidence equal, a one unit increase in decision confidence was associated with a .63 point increase in post-choice regret.

To test our final hypothesis, we are interested in whether or not significant differences in our variables of interest can be attributed to the PES interface. For this purpose, a t-test provides a suitable method. Keeping with common practice, we set a significant level of 0.05 to test our hypotheses.

There was a significant observed difference for product choice confidence among users of PES1 and PES2 (t value -2.77; p-value=0.0065). A significant difference was also observed for decision confidence scores for PES1 and PES2 users (t value -3.62; p-value=0.0004). Finally, A significant difference in post-choice regret was also found (t value -3.55; p-value=0.0005). Table 3 summarizes the results of the hypotheses testing.

4 Discussion and Future Research Directions

In this study we examined the way in which PES interface design choices either promote or detract from the usefulness of PES content. Hypotheses H1, H2, and H3 were all supported by the data. This provides support for the overall research question of whether PES can influence consumer confidence in the online decision making process by lowering feelings of both post-choice regret and its antecedents. As expected, individuals with higher feelings of confidence in both their product choice and their decision process considered themselves less likely to experience post-choice regret at a future point in time.

The second question of interest in this study is whether or not these benefits of PES can be attributed to specific design characteristics, namely a PES ability to summarize online customer reviews in a relevant and meaningful way. Support was found for Hypotheses H3a, H3b and H3c. This provides some statistical evidence that PES design characteristics do impact the consumer decision making process to varying degrees.

Table 3. Hypothesis Test Results

Hypothesis/Finding	p-value	Supported/ Not Supported
H1: Product choice confidence in this analysis had a significiant impact on user perceptions of post-choice regret	p-value=0.007	Supported
H2: Decision confidence in this analysis had a significant impact on user perceptions of post-choice regret	p-value=.0001	Supported
H3a: Statistically significant product choice confidence differences were found when comparing PES1 users to those of PES2.	p-value=.0065	Supported
H3b: Statistically significant differences were found for decision confidence when comparing PES1 users to those of PES2.	p-value=.0004	Supported
H3c: Statistically significant differences were found for perceptions of post-choice regret when comparing PES1 users to those of PES2	p-value=.0005	Supported

Interestingly, PES 1 outperformed PES 2 in every category. From the standpoint purely of information overload, we could expect PES 2, with its summary features and ability to extract relevant data from large masses of reviews to have a larger benefit on information quality. However, participants found that the online review content plays a useful role in the online decision making process, but that the characteristics of PES design ultimately determine both how much and the nature of the benefit provided by the system. PES 1, with its unfiltered presentation of online customer reviews, outperformed PES 2's filtered summary data in every category including product and decision confidence, as well as perceived post-choice regret.

One limitation of the study is concerns the comparison of PES1, which is widespread and established, with the lesser known PES2 system. It is worth noting therefore that this widespread familiarity with the PES1 interface may have impacted the responses of survey participants. In terms of name recognition, however, Google is also a very established brand, and therefore should eliminate most of the risk associated with extraneous variables such as trust or user self-efficacy.

Another possible explanation for the higher scores of PES1 may lie in the nature of the review presentation. PES1, in preserving the narrative context of the online customer reviews, could be providing consumers with a product that is greater than the sum of its parts. This sentiment was reflected in some of the qualitative responses that we collected. When asked what they liked about the system, users had this to say.

- *(PES 1) had information on other things that the users buying this product had looked at and/or chosen. The descriptions of the products were easily accessible and detailed.*
- *(PES 1) has long and comprehensive user reviews. Also, the summaries are more comprehensive and informative.*

- *I like the way that (PES1) had ratings that were in-depth and beyond an overall product rating.*

Many users of PES2 did report positive aspects of the interface design. Most often, these were related to the more fine-grained control that PES2 gave of the viewer environment. In addition to summary data, PES2 also allowed users to specify the nature of the keywords used in summarizing relevant reviews. Several users found this particular feature very helpful, saying that Google Products allowed control over the viewing of reviews, and presented information in a concise and clear format.

In summary, while PES1 outperformed PES2 in this analysis, there are numerous takeaways from this study that can be used to design a future artifact encompassing the best aspects of both systems. Additionally, in formalizing and testing the relationship between PES design and important decision-making constructs such as perceived post-choice regret, the study makes a significant contribution to both theory and future design science practice.

References

1. Keller, K.L., Staelin, R.: Effects of quality and quantity of information on decision effectiveness. Journal of Consumer Research 14(2), 200–213 (1987)
2. Ho, J., Tang, R.: Towards an optimal resolution to information overload: an infomediary approach. In: Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work, pp. 91–96. ACM, New York (2001)
3. Liang, T.P., Kai, H.J., Ku, Y.C.: Personalized content recommendation and user satisfaction: Theoretical synthesis and empirical findings. Journal of Management Information System 23(3), 45–70 (2007)
4. Ng, V.W.K., Chyi, H.I.: The Effects of Context Clutter and Advertising Repetition on Attitudinal and Behavioral Changes toward an Online Advertisement. Embedding Into Our Lives: New Opportunities and Challenges of the Internet 145 (2008)
5. Mudambi, S.M., Schuff, D.: What makes a helpful Online Review? A study of customer reviews on Amazon.com. MIS Quarterly 34(1), 185–200 (2010)
6. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 168–177. ACM, New York (2004)
7. Hu, N., Pavlou, P., Zhang, J.: Can online reviews reveal a products true quality. In: Proceedings of ACM Conference on Electronic Commerce (EC 2006) (2006)
8. Chevalier, J.A., Mayzlin, D.: The effect of word of mouth on sales: Online book reviews. Journal of Marketing Research 43(3), 345–354 (2006)
9. Bhattacharjee, S., Gopal, R.D., Lertwachara, K., Marsden, J.R.: Consumer search and retailer strategies in the presence of online music sharing. Journal of Management Information Systems 23(1), 129–159 (2006)
10. Loomes, G., Sugden, R.: Regret theory: An alternative theory of rational choice under uncertainty. The Economic Journal 92(368), 805–824 (1982)
11. Bell, D.E.: Regret in decision making under uncertainty. Operations Research 30(5), 961–981 (1982)
12. Tsiros, M., Mittal, V.: Regret: A model of its antecedents and consequences in consumer decision making. Journal of Consumer Research 26(4), 401–417 (2000)

13. Zeelenberg, M.: Anticipated regret, expected feedback and behavioral decision making. Journal of Behavioral Decision Making 12(2), 93–106 (1999)
14. Das, N., Kerr, A.H.: "Woulda, Coulda, Shoulda": A Conceptual Examination of the Sources of Post-choice Regret. The Journal of Marketing Theory and Practice 18(2), 171–180 (2010)
15. Hung, S.Y., Ku, Y.C., Liang, T.P., Lee, C.J.: Regret avoidance as a measure of DSS success: An exploratory study. Decision Support System 42(4), 2093–2106 (2007)
16. Baskerville, R., Pries-Heje, J., Ramesh, B.: The enduring contradictions of new software development approaches: a response to Persistent Problems and Practices in ISD. Information Systems Journal 17(3), 241–245 (2007)
17. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. Mis Quarterly 28(1), 75–105 (2004)
18. Pries-Heje, J., Baskerville, R., Venable, J.: Strategies for design science research evaluation. In: Proceedings of the 16th European Conference on Information Systems, pp. 255–266 (2004)
19. Kumar, N., Benbasat, I.: Research Note-The Influence of Recommendations and Consumer Reviews on Evaluations of Websites. Information Systems Research 17(4), 425 (2006)
20. McKinney, V., Yoon, et al.: The measurement of web-customer satisfaction: An expectation and disconfirmation approach. Information Systems Research 13(3), 296 (2002)
21. Simonson, I.: The influence of anticipating regret and responsibility on purchase decisions. Journal of Consumer Research 19(1), 105–118 (1992)
22. Tan, C.H., Yang, X., Teo, H.H., Lin, G.: An Empirical Investigation of the Auction Buyer's Choice to Buy Out or Bid: Cry of Regret or Laugh of Satisfaction? In: Proceedings of ICIS 2005, vol. 62 (2005)
23. Qin, L., Gulati, A., He, L., Fang, W.: Adverse Selection for Luxury Goods in Online Auctions. Journal of Internet Commerce 8(3), 268–287 (2009)
24. Dave, K., Lawrence, S., Pennock, D.M.: Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. Proceedings of the 12th international conference on World Wide Web. ACM, 528 (2003)
25. Kwaifunip, R., Wagner, C.: Weblogging: A study of social computing and its impact on organizations. Decision Support Systems (45), 242–250 (2008)
26. Jiang, Z.: The effects of presentation formats and task complexity on online consumers. MIS Quarterly (2007)

Managing the Future—Six Guidelines for Designing Environmental Scanning Systems

Jörg H. Mayer

University of St.Gallen, Institute of Information Management,
Mueller-Friedberg-Straße 8, 9000 St. Gallen, Switzerland
joerg.mayer@unisg.ch

Abstract. The 2008/2009 economic crisis provided a sustainable impulse for improving environmental scanning systems. Although a substantial body of knowledge exists, these concepts are not often used by practitioners. Based on a literature research, this article outlines six guidelines for designing environmental scanning systems that are more applicable than the state of the art. We incorporate these guidelines in a six-step method and focus on how the capabilities of "modern" information systems (IS) enable a better "grasp" of weak signals and a closer incorporation of the findings into the executives' decision-making process. Applying this reworked method at a raw materials and engineering company leads to a first instantiation—the "Corporate Radar." The version discussed here ends with a value-driver tree showing economic value added at risk on top. The resulting lessons learned helped us in two ways: providing concrete starting points for future research and arguing for the research method presented here.

Keywords: Corporate management, balanced chance and risk management, corporate business intelligence (BI), method design.

1 Introduction

The increasing volatility of their companies' environment is a growing concern for executives. In particular, they worry about not being prepared for environmental shifts or—even worse—not being able to parry them.

The 2008/2009 economic crisis gave a sustainable impulse for focusing earlier on emerging threats and opportunities [1, 2]. Environmental scanning—ideally, IT-based within a corporate business intelligence (BI) architecture[1]—can help to manage this challenge. Its main function is to gather and interpret pertinent information that would assist management in planning the organization's future course of action [4, 5]. Companies that do so will have brighter prospects than those that do not [6].

In literature a substantial body of knowledge exists, but it often goes *unused* in practice [7-9]. Practitioners perceive the task as a difficult one per se. Some may not even know how to start, experiencing difficulties in design, implementation, and

---

[1] BI is a broad category of technologies, applications, and processes for gathering, storing, accessing, and analyzing data to help its users make better decisions [3].

H. Jain, A.P. Sinha, and P. Vitharana (Eds.): DESRIST 2011, LNCS 6629, pp. 276–290, 2011.

day-to-day operation [10]. The objective of this article is therefore to design environmental scanning systems that are more applicable than the state of the art.

This article adheres to design science research (DSR) in information systems (IS)—a discipline that focuses on developing innovative, generic solutions for practical problems and thus emphasizes utility [11]. Design science (DS) distinguishes four types of artefacts: constructs, models, methods, and instantiations [12]. We set out a method. While they can range from algorithms to practical instantiations [11], we present six guidelines which recommend how the design of environmental scanning systems should be reworked.

Various processes exist for developing artefacts under the DS paradigm [11-13]. The one described by March & Smith [12], which distinguishes between "build" and "evaluate," predominates in the literature [11]. Emphasizing these activities, we apply the research process model of Peffers et al. [13] with six phases: *Identify a problem and motivate the research:* We start the article with lessons learned from the 2008/2009 economic crisis. Reworked environmental scanning systems should help executives to focus earlier on emerging threats and opportunities. *Define objectives of an (IS) solution*: After revisiting relevant foundations (Sec. 2) and performing a literature review, we identify gaps between the body of knowledge and their infrequent use in practice (Sec. 3). *Design and develop*: As designing environmental scanning systems is not a greenfield endeavor, we give a structure to existing recommendations and add new ones to overcome the identified gaps (Sec. 4). *Demonstrate*: We then apply the reworked method at a raw materials and engineering company with the result of a "Corporate Radar" instantiation (Sec. 5.1). *Evaluate*: That prototype provides a direct feedback on the methods' outcome and we discuss the lessons learned (Sec. 5.2). They help us to argue for the research method presented here (Sec. 6). *Communicate*: The key findings should also provide concrete starting points for future research (Sec. 7).

2 Foundations

A company's environment is defined as the relevant physical and social factors within and beyond the organization's boundary [14]. While operational analysis most often focuses on (short-term) internal difficulties in the implementation of strategic programs [15], the scope of *strategic* environmental scanning is to anticipate (long-term) environmental shifts and to analyze their potential impact. This article concentrates on the latter, hereafter referred to as "environmental scanning." As strategic issues can emerge within or outside a company, changes in both, a companies' external and internal environment are reflected [16].

Management support systems (MSS) are proposed as a label for IS to support for managerial decision making. They cover decision support systems, management information systems, executive information systems (EIS), knowledge management systems, and business intelligence [19]. In this domain, executive information systems (EIS) focus on top management. *Environmental scanning systems* in turn have their roots in management literature [4] focusing on the executives' task to be aware of environmental trends [17, 20]. So they specify the sectors to be scanned, monitor the most important indicators that may create opportunities or threats for the company, cover the (IT-based) tools to be used [17, 18], and often assign responsibilities to support environmental scanning [5]. The latter is not covered in this article.

Two modes of information collection are distinguished [21]: Besides the reactive mode, in which information is acquired to resolve a problem, we follow the proactive mode, in which the environment is scanned for upcoming changes representing opportunities and threats.

3 Current Need for Environmental Scanning Systems

3.1 Regulatory Needs

Environmental scanning is not just "nice to have," as Kajüter [22] shows in his multi-country comparison. In the wake of several cases of fraud around the turn of the millenium, legislators expressed a need for a more detailed approach to risk management. Best known is the U.S. Sarbanes-Oxley Act. In particular, Section 404 requires companies listed on the New York Stock Exchange to extensively document internal controls, establish independent audit committees, and it made audits of internal controls mandatory [23].

Furthermore, "going concern" is no longer a voluntary statement. Since 1998, both the International Financial Reporting Standards (IFRS) framework and the IAS 1 specify that audits must incorporate the "foreseeable future" of at least one year. Environmental scanning systems should be integrated into such forward-looking risk management [24].

3.2 State of the Art

We performed a literature review [25] combining elements of IS design theories along with the research method used (Fig. 1): Requirements are prerequisites, conditions, or capabilities needed by the users of software systems [26]. Models outline concrete systems, features, or combinations of these [27]. Methods cover the environmental scanning processes employed. Papers are regarded as empirical approaches if they rely on observation or other empirical methods. Design approaches involve ideas and frameworks for creating a better world.

Our literature search followed vom Brocke et al.'s [28] four-step process: First, using the MIS Journal Ranking [29], we selected six of the most popular IS journals[2] [25]. We also looked at A-ranked conference proceedings[3]. Second, we used EBSCO host, Google scholar, Science Direct, and Wiley Inter Science to access the journals. Third, using the keywords "environmental scanning system" and "early warning system, weak signal, leading indicator" they produced 14 relevant hits in total. Fourth, doing a backward and forward search, we end with 70 relevant publications (Fig. 1).

[2] MIS Quarterly (MISQ), Decision Support Systems (DSS), Information & Management (I&M), Journal of Management Information Systems (JMIS), European Journal of Information Systems (EJIS), and Information System Management (ISM).

[3] International and European Conferences on Information Systems (ICIS, ECIS).

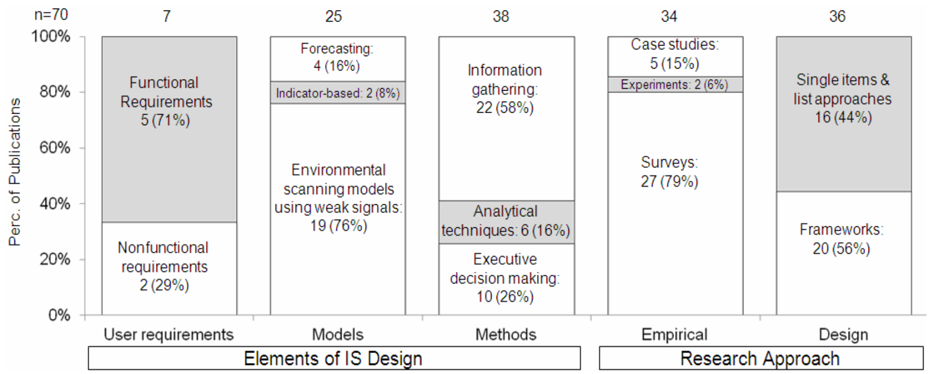


Fig. 1. Classification of the publications

Lack of research on sound requirements analysis: Looking at Fig. 1, the first apparent issue is that just seven out of 70 publications focus on requirements. To develop a corporate radar, Xu et al. [30] conducted a cross-industry study and found that task-related environmental areas are perceived as more important than far-general information is. Another four studies reviewed the functional requirements of executives in terms of their scanning practices. For example, Yasai-Ardenaki & Nystrom [18] emphasize a link between scanning areas and strategy. Especially in volatile industries, the "periphery"—a metaphor describing latent changes that can result from, e.g. political movements—should be considered in environmental scanning [9]. Regarding Daft et al. [31], at a minimum the periphery should cover science, politics, law and justice, and international relationships. Focusing on the non-functional perspective, El Sawy [32] suggests that a successful solution should have just a limited number of scanning areas and sources.

Existing models lack the "grasp" to apply weak signals in practice: Out of our 70 articles, 25 publications present models for environmental scanning systems. These can be divided into three generations: Early-warning models were first mentioned in the late 1960s [33]. Forecasting advances this technique using time series, not only for actual and planned data ("as-is/to-be comparison"), but also planned and extrapolated as-is data. Second-generation scanning systems identify latent risks and chances. Such an indicator-based model is described by Davies et al. [15]. They recommend standardized indicators and provide guidelines for selecting them. Reference values and ranges of tolerance are employed to avoid overreactions to random fluctuations.

Finally, Ansoff [6] introduced the concept of weak signals. An example is the spread of carbon fibre. Its usage has increased steadily in recent years, and its potential to serve as a substitute for steel represents a strategic issue. Ansoff's concept remains topical in recent literature. 76 percent of the publications about models we researched use his approach (Fig. 1): Narchal et al. [17] promise that their systematic scanning and monitoring is more effective than ad hoc scanning concerning forecasting the impact of signals on the organization. Their approach uses so-called descriptors. Examples are general elections, which bundle many "soft factors" that can affect the organization. From our instantiation we learned that carbon-trading is such a factor determined by elections. Therefore changes of government indicate threats or opportunities. The body of knowledge covers several of such models but they lack applicable guidelines.

Lack of methods to incorporate environmental scanning results into executives' decision making: Aguilar [4] examined modes of scanning, namely undirected viewing, conditional viewing, and informal and formal search. Attaining strategic advantages through information gathering is a subject of high interest. 58 percent of articles on methods refine the concept.

Two types of analytical techniques can be distinguished. The first employs mathematical methods often used by banks. But, the 2008/2009 economic crisis showed that such methods have significant shortcomings. Complicated premises [2], the fact that using confidence intervals leads to the exclusion of high-impact events, and the devastating effect small errors in assumptions can have on the outcome make this type of technique unsuitable for application [34]. The alternative are heuristic techniques [16]. For example, the Delphi method involves questioning experts, following certain rules in order to avoid a majority opinion too early [35]. Narchal et al. [17] recommend using influence diagrams focusing on levers and their influence on important environmental descriptors. In order to model dependencies between single items, cross-impact matrices evolved [35]. They can even contribute finding the most probable future scenario. Articles dealing with executive scanning behavior do not provide concrete guidelines on how to design environmental scanning systems [36].

3.3 Empirical Facts

Fuld [7] showed that 97 percent of US companies lack an early warning system. Interviews with 140 corporate strategists found that two-thirds had been surprised by as many as three high-impact competitive events in the past five years. Following Krystek & Herzhoff [8], 30 percent of European chemical companies do not have environmental scanning systems in place. Furthermore, 15 percent said the instruments available are not accepted for practical use.

Similar findings are reported from companies listed in the Financial Times "Europe 500" report [37]: Most of the executives consider scanning concepts as too complex and difficult to implement. Therefore, the results are not a substantial part of their decision-making.

In the light of previous considerations, a gap between literature and practice becomes obvious. Despite the clear benefits environmental scanning systems offer, companies do not use them. In particular, a more applicable method is lacking to fulfill the business requirements we identified. To overcome this gap, we focus on addressing the following drawbacks: first, the "grasp" of weak signals needs improvement. Thus, introducing scanning results into executives' decision-making process will require methodological support; and second, mathematical models are too complex.

4 Method Design

For the method design we start with first ideas from Narchal et al. [17] and Mayer & Wurl [38]. The latter refer to their approach as Corporate Radar. Based on the findings of our literature analysis, we incorporate six guidelines that make the IS design method more applicable than the state of the art is.

4.1. Take a 360-Degree Approach, But Select Just the Most Important Environmental Scanning Areas

When designing environmental scanning systems, we recommend prioritizing task-related areas for executives [30]. Because they have to manage the company, the radar must provide a "360 degree" view (Fig. 2), but only the most important areas should be selected for scanning [32]. More specifically, the design of environmental scanning systems should start by reflecting the company's vision and strategic program [18].

The task-related environment consists of suppliers, customers, and competitors. The more volatile the environment is, the more the general environment should be scanned as well [9, 31]. It can be described using the PESTL scheme [31].

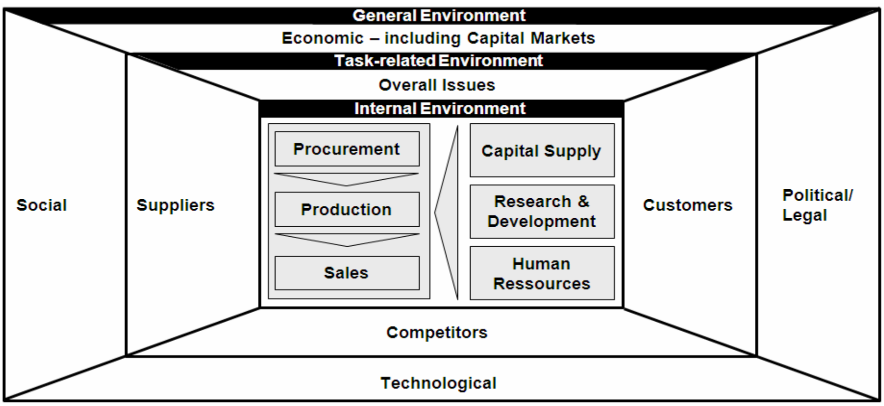


Fig. 2. Scanning areas for reworked environmental scanning systems

To select internal scanning areas, we follow the value chain, which consists in the industrial sector of procurement, production, and sales (a more complex model is presented in [39]). In terms of supporting areas, strategic issues often emerge in human resources, research & development, and capital supply. These can involve, e.g., employee satisfaction or the development of a new technology [16]. This first activity in setting up a Corporate Radar does not necessarily require IT support.

4.2 Define Concrete Descriptors and Use IT to Proactively Identify Relevant Cause-and-Effect Chains

To anticipate future shifts, we adopt the idea of descriptors. But to reduce the equivocality of scanning findings [31] also indicators are subsumed under descriptors. We see two levers to improve the "grasp" of weak signals. The first is to distinguish relevant descriptors from the bulk of data. Therefore, evaluating them in terms of lead time, clarity and appropriate cost/impact ratio is proposed. For example, the Baltic Dry Index provides a concrete descriptor for the sales area that meets the above-mentioned requirements. It is the "visible end" of a cause-and-effect chain measuring the rates charged for chartering dry bulk vessels to indicate economic development.

The second lever is to identify patterns to foresee future developments. Artificial neural networks, data mining, and semantic search, compared to precedent techniques or a human observer, might improve processing weak signals and extracting cause-and-effect chains.

4.3 Leverage IT to Automate Day-to-Day Routines and to Follow the Descriptors' Movements

In a third step, data sources for the descriptors and the frequency of data collection must be determined. A trade-off is necessary between the cost of data collection, such as license fees of data sources, costs of additional employees, the reporting system itself, and its ability to indicate potential opportunities and threats.

We emphasize the Internet [9] as well as capital markets [36] as data sources, which provide useful, condensed information on competitors, suppliers, and customers. Using supportive, predefined user interfaces for data access or common IT languages, such as XBRL (eXtensible Business Reporting Language), facilitating to access these information sources by automated routines to systematically monitor the movements of the descriptors. To save even more cost and time to process information Frolick et al. [40] propose gathering data through computerized notes.

4.4 Translate Descriptors' Impact into a Balanced Opportunity-and-Threat Portfolio and Leverage Expert Experience Documented in an Impact-Matrix

Once descriptors and their data sources have been defined, a model of their impact on company performance must be created. Therefore critical assumptions, which must hold, if a company is to achieve its objectives are in focus. To do so, we first designate ranges of tolerance and then apply a heuristic analysis. Using the basic functionality of cross-impact matrices, we propose the *impact-matrix* (Fig. 3) to model dependencies and to define company-specific opportunities and threats. Thus, the matrix must cover two types of variables: The columns show the descriptors derived from the scanning areas and the rows contain assumptions that serve as the basis for scenarios. The rows should also cover constitutive risks—in other words, risks that cannot be affected by the company, e.g. plagiarism. To measure the impact descriptors have on the assumptions, we apply a one-dimensional scoring to conjugate their future state.

If a critical value for an assumption is exceeded (see assumption 2, Fig. 3), an alert is triggered, signaling either a threat or an opportunity. Large deviations in the values of single descriptors or in assumptions must be reported ad hoc as "breaking news" or "turning points" that refute prior assumptions. Typical assumptions are about market growth or the economic development of important customers.

An interpreted *descriptor map* allows quantifying threats and opportunities (Fig. 4, left hand side). The x-axis shows the score from the impact matrix and the y-axis the estimated lead time. Critical descriptors are bundled for the assumption to which they refer and are highlighted according to their direction of development. Depending on descriptors' development and score, either a threat or an opportunity alert is signaled.

Scores: -2 - strongly negative; -1 - negative; 0 - no correlation; 1 - positive; 2 - strongly positive

Scanning Area	Suppliers			Buyers			...	Political/Legal		Sum**	Alert Oppor-tunity*	Alert Threat*
Descriptor	No.	Quality	...	No.	Solvency	...	...	Elections	...			
critical	above	below	...	not critical	not critical	...	...	not critical	...			
Assumption 1	1	0	...	2	0	...	...	1	...	4	25%	0%
Assumption 2	1	2	...	0	-2	...	...	1	...	6	17%	33%
...	...	...	...	...	...	...	...	...	...	...	...	...
Assumption n	2	-1	...	2	0	...	...	0	...	5	40%	20%
Sum**	4	3	...	4	2	...	...	2	...		* threshold = 30% ** absolute values	

Fig. 3. Impact-Matrix

To incorporate the findings into the executives' decision-making process, we transfer them into the *balanced opportunity-and-threat portfolio* (Fig. 4, right hand side) for a condensed presentation. The number of descriptors, bundled by assumption, helps to quantify the potential impact. The score from the impact matrix helps to quantify the likelihood of occurrence and the descriptors' typical lead time helps to quantify the time to impact. Applying this method enables corporations to comprehend short-term change using human experience rather than mathematical models.

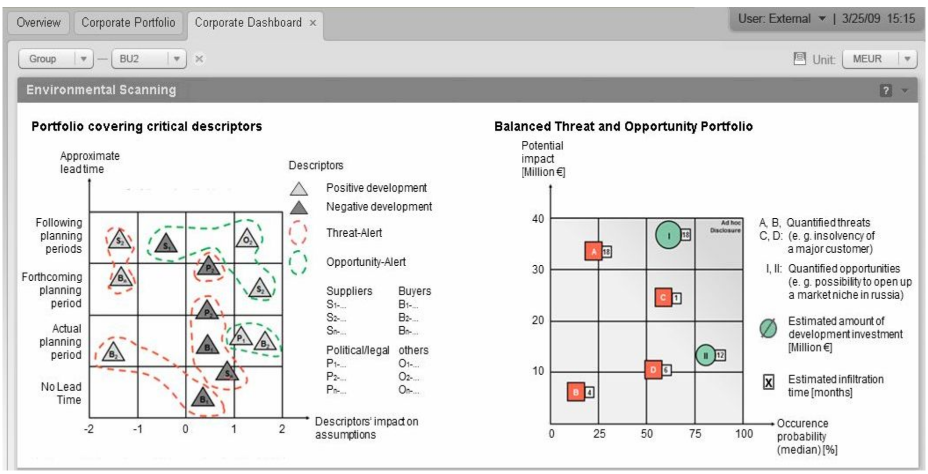


Fig. 4. Balanced opportunity-and-threat portfolio (screen shot from the instantiation)

4.5 Incorporate Scanning Results into Executives' Decision-Making Process by Generating Scenarios from the Set of Environment Assumptions

The balanced opportunity-and-threat portfolio is used to predict those aspects of the future that are relevant to the company. To ensure that executives receive these findings in an amount and form that facilitate effective decision making, we link the identified opportunities and threats with the companies' management control approach [16, 40]. In our instantiation, this took the form of a value-driver tree representing economic value added at risk (Fig. 5). Once the descriptor values and the associated opportunities and threats have been determined, they should be used to define three scenarios: optimistic, most probable, and pessimistic [17].

4.6 Use Retrospective Controls to Continuously Update the IS and Collaboration to Share the Scanning Findings in Day-to-Day Work

New findings should be used to verify the assumptions and determine whether the method applied requires modification. To monitor the need for new descriptors, checking the lines' and columns' total of the impact-matrix absolute values (Fig. 3) is helpful. In addition to ad hoc reporting (Sec. 4.2), a periodical presentation of findings helps to identify the most probable scenario and critical assumptions for the next planning period. Such reviews ensure that recognized issues are monitored and adding descriptors further explains identified chances and risks. Furthermore, the disappearance of opportunities and threats can be displayed. Groupware allows e-mailing and other forms of collaboration [41].

5 Demonstrate

An instantiation at a large, international company in the raw material and engineering sector (Europe, sales: USD 56 bn; employees: 174,000) helped us to evaluate the reworked method on hand and to make the guidelines more concrete.

5.1 Objective, Results, and Future Design

During the 2008/2009 economic crisis, both, the business side and IT department recognized that they could contribute better to executives' decision making with information about potential opportunities and threats. A 360-degree environmental scanning system was required following two main objectives: (1) Enhanced analysis should provide more information on company's environment—going beyond standard business parameters to analyze events, trends, and forces that could radically alter the future of the company. (2) To incorporate the findings into executives' decision making, an integration of the scanning findings into group reporting was mandatory.

The CIO applied our reworked six-step method. First, the areas for environmental scanning were delineated. Then three to five descriptors were selected for each scanning area. Data sources were identified and their update frequency was determined. In a fourth step, tolerance limits were defined to avoid overreactions and assumptions were drawn to be linked with the descriptors by questioning internal professionals ending in the impact matrix (Fig. 3). The descriptor map allows synthesizing the company specific balanced opportunity-and-threat portfolio. It integrates the findings of the scanning process into executives' decision-making process (Fig. 4). The radar discussed here culminates in an IT-based value-driver tree showing economic value added at risk on top (Fig. 5).

A new business application with a user-friendly user interface ("frontend") provides the scenario visualization that allows switching between the best, worst and most probable scenarios (Fig. 5, right hand side). Best and worst case scenarios define the range of the most important value drivers such as net sales and costs. Because of the mathematical connections between them, ranges for the financial performance indicators EBIT, ROCE and EVA (Fig. 5, left hand side) are defined as well. The slider position represents the most probable scenario. All drivers can be moved to the right or to the left to simulate changes no matter which scenario is selected.

The Corporate Radar is implemented in a four-layer architecture. It uses a data warehouse to extract and transform data from the various transaction systems and stores it in a central location. In our case it is a SAP Business Warehouse. Query technologies carry over the IT restrictions based on separate data storages. A planning, consolidation, and strategy management application, SAP BO Planning and Consolidation and SAP Business Objects Strategy Management, accesses the environmental scanning data and incorporates it into the executives' decision-making process by providing management reports with a new governance, risk, and compliance information cluster. The data is presented in the uppermost (fourth) layer. We used SAP BO Xcelsius as the frontend.

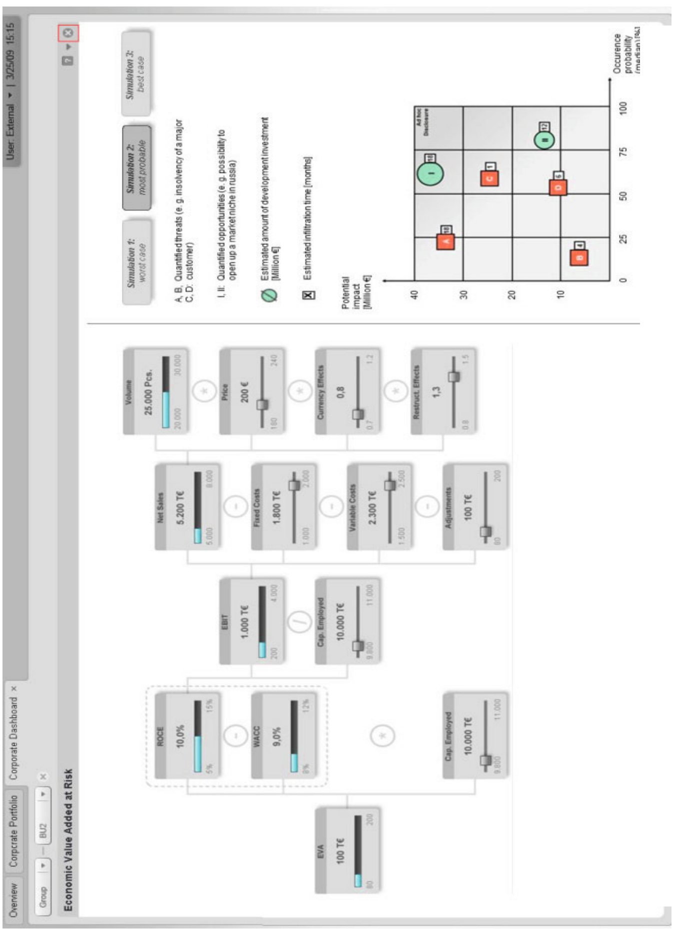


Fig. 5. Incorporating the results of environmental scanning systems into executives' decision making by scenario technique (screen shot)

Four months after introducing the prototype, we researched the day-to-day experience to see how the instantiation goes on, especially how to identify the descriptors leveraging "modern" IT capabilities. This effort was conducted both internally and externally using free data from the Web and the help of market research firms. In particular, the CIO is interested in using semantic search in future to go more deeply beyond the standard business parameters.

5.2. Lessons Learned from the Instantiation

The case example, along with three interviews[4] in other industries, shows that one of the success factors of such IS design methods is direct interaction with executives to overcome their overwhelming business requirements.

Lesson 1: The design of environmental scanning systems is predominantly a top-down business project, and it should be communicated that way. One challenge—which arose unexpectedly right at the beginning of the project—was to align executives' individual perspectives with company's existing vision and strategic program (Sec. 4.1). Our information-needs analysis thus involved jointly rethinking the company's strategic program, as well as how descriptors can make an actionable contribution proactive to its goals. A broad range of diffuse forward-looking information was available that needed to be streamlined.

Lesson 2: Project managers can establish their role in environmental scanning systems, if they can act as mediators between business requirements and IT capabilities. The most important analyses were useful in achieving a second type of alignment: one between business requirements and increasing IS capabilities. By connecting the right parties across the company, the CIO brought in an interdisciplinary business/IT perspective that makes him the ideal mediator in the course of the design.

Lesson 3: Executives must champion the complete IS development process, devoting time not just to the analysis of their information needs, but to reviewing the prototype as well. Besides participating in the information needs analysis, executives must take an active role in reviewing the prototype as well—an area in which they are, for the most part, unfamiliar. To do so, they are best served by providing them with hands-on "clickable" prototypes and then making enhancements step by step.

In terms of the method itself, another three lessons in comparison to other holistic approaches such as Narchal et al. [17] or Frolick [40] emerge as follows:

Lesson 4: The requirements analysis should be transparent right from the beginning: Structured interviews in method phases one to three (Sec. 4.1-4.3) helped to involve executives, risk managers, accountants, and IT experts from the beginning of the project. Such a feedback culture enabled transparency regarding how findings are generated and used in the IS design—a solid basis for its later acceptance by all the stakeholder groups. Herein, the impact matrix (Fig. 3) shows directly the interdependencies between the descriptors (and its changes) and the assumptions for the

[4] Chemicals: Europe, employees about 105,000, sales about USD 65 billion; high-tech: Asia and North America, 35,000, USD 15 billion; automotive supply: Europe, 150,000, USD 32 billion.

scenarios at a glance enabling the proposed better "grasp" of weak signals. Data mining and semantic search can accomplish the task of identifying the descriptors, but they are not easy to implement and handle. A better first step are *scanning routines* with alert functionality implemented to automate basic tasks such as insolvency scans of the most important customers or significant changes in daily currency rates.

Lesson 5: Design for use: We skipped the mathematical models and reworked the method on hand using heuristic techniques. Especially the *balanced opportunity-and-threat portfolio* synthesizes the findings of environmental scanning this way. To integrate the findings into executives' decision-making process, we linked the results with the economic value added, often used in practice, and expanded their key figures with a "at risk" perspective. Furthermore, we provided an ease of use IS to simulate the impact of environmental scanning with sliders on the scenarios defined before. This helped to overcome the researched gaps that premises of mathematical models and confidence intervals are not easy to understand.

Lesson 6: Provide adaptability for future enhancements within a business/IT approach: A modular design allowed us to improve the Corporate Radar step by step, from the Delphi method, vis-á-vis the impact matrix, the portfolio to the first prototype. Furthermore, some work—e.g., the analysis of the descriptors' data sources and the design of the prototype—could be done in parallel, making it possible to compensate for project delays. Herein, it was useful to take a business/IT architecture of the four layers "strategy, organization, alignment, and IT support" [36] not only helping us to streamline the objective of the companies' Corporate Radar, but also to address the IT-architecture aspects of the implementation.

6 Evaluation of the Research Method on Hand

Comparing the findings from the instantiations with the state of the art reveals some points worthy of discussion. First, the method on hand offers greater rigor than action research does for several reasons. *Up-stream compatibility*: A literature analysis as a starting point not only makes it possible to identify current design gaps, but even to leverage the existing body of knowledge. Incorporating some publications from practitioners would expand this knowledge with insights from day-to-day operations. *Distinctiveness*: A literature review leads to guidelines that are nearly complete and certainly distinct. Using a framework to structure the literature analysis aids basic orientation. *Downstream compatibility*: The instantiation provided an opportunity for a first evaluation of the utility of the guidelines we derived. For future research, modular guidelines can be pursued rather than, for example, structural models.

Second, as IS research is losing relevance in practice [43], the advantages of this method over "general" empirical surveys and structural models can be specified as follows: *Traceability*: Combining literature research with "build, demonstrate, and evaluate" activities offers rigor without losing the key DSR objective: to build new artefacts to create a better world. In doing so, an instantiation helped us not only to evaluate the utility of the guidelines synthesized from the literature review, but also to specify them in practice. This should increase practitioners' acceptance of the work. *Handling and direct use of findings*: Guidelines should be easier to work with than conceptual findings of surveys or hypotheses of structural models. They can be used one by one not only to define reworked IS, but also to evaluate existing IS as well.

7 Outlook and Future Research

The objective of this article was to set out guidelines to rework environmental scanning systems. We derived them from the findings of a literature analysis and integrated them in a six-step IS design method. Thereby, we highlight consistent IT-based scanning and processing of descriptors to improve the "grasp" of weak signals. To better incorporate the scanning findings into executives' decision-making process, we leveraged an IT-based opportunity-and-threat portfolio. Applying the reworked method at a raw materials and engineering company brings up a "Corporate Radar," ending with a value-driver tree showing economic value added at risk on top along with capabilities to simulate scenarios and their descriptors' impact on the most important financial KPIs.

So far, our research has been limited to a restricted number of publications. We covered the leading journals, but we see future research to expand this coverage, especially by identifying more practitioner publications that do not include any of our search terms. However, this should be a secondary issue.

Focusing on the DSR perspective, it is more important to do additional case studies to specify the guidelines discussed here with "demonstrate" and "evaluate" activities. Another contribution could be a survey to get a direct perspective on executives' requirements for environmental scanning systems in a broader sample.

Due to the number of instantiations so far it is not possible to determine for sure whether these guidelines lead to more applicable environmental scanning systems. At least they provide concrete starting points for future research. Our own research will use additional instantiations to determine the extent to which these guidelines and, hopefully, forthcoming extensions are generalizable. Moreover, an empirical study is underway to identify user types, use cases, and IS access modes for a more situational approach of environmental scanning systems to better meet executives' individual needs.

References

1. Hopwood, A.G.: The economic crisis and accounting: Implications for the research community. Accounting, Organizations & Society 34, 797–802 (2009)
2. Makridakis, S., Hogarth, R.M., Gaba, A.: Why Forecasts Fail. What to Do Instead. MIT Sloan Management Review 51, 83–90 (2010)
3. Wixom, B.H., Watson, H.J.: The BI-based organization. International Journal of Business Intelligence 1, 13–28 (2010)
4. Aguilar, F.J.: Scanning the Business Environment. Macmillan, New York (1967)
5. Lenz, R.T., Engledow, J.L.: Environmental Analysis Units and Strategic Decision-making: a Field Study of Selected “Leading-edge” Corporations. Strategic Management Journal 7, 69–89 (1986)
6. Ansoff, H.I.: Managing Strategic Surprise by Response to Weak Signals. California Management Review 18, 21–32 (1975)
7. Fuld, L.: Be Prepared. Harvard Business Review 81, 20–21 (2003)

8. Krystek, U., Herzhoff, M.: Szenario-Technik und Frühaufklärung: Anwendungsstand und Integrationspotential. Zeitschrift für Controlling und Management 50, 305–310 (2006)
9. Day, G.S., Schoemaker, P.J.H.: Scanning The Periphery. Harvard Business Review 83, 135–148 (2005)
10. Lesca, N., Caron-Fasan, M.-L.: Strategic scanning project failure and abandonment factors: Lessons learned. European Journal of Information Systems 17, 371–386 (2008)
11. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28, 75–105 (2004)
12. March, S.T., Smith, G.F.: Design and Natural Science Research on Information Technology. Decision Support Systems 15, 251–266 (1995)
13. Peffers, K., Tuunanen, T., Gengler, C.E., Rossi, M., Hui, W., Virtanen, V., Bragge, J.: The Design Science Research Process: A Model for Producing and Presenting Information Systems Research. In: DESRIST, Claremont, CA, pp. 83–106 (2006)
14. Duncan, R.B.: Characteristics of Organizational Environments and Perceived Environmental Uncertainty. Administrative Science Quarterly 17, 313–327 (1972)
15. Davies, J., Finlay, M., McLenaghen, T., Wilson, D.: Key Risk Indicators – Their Role in Operational Risk Management and Measurement. In: ARM and RiskBusiness International, Prague, pp. 1–32 (2006)
16. Ansoff, H.I.: Strategic Issue Management. Strategic Management Journal 1, 131–148 (1980)
17. Narchal, R.M., Kittappa, K., Bhattacharya, P.: An Environmental Scanning System for Business Planning. Long Range Planning 20, 96–105 (1987)
18. Yasai-Ardenaki, M., Nystrom, P.C.: Designs for Environmental Scanning Systems: Tests of a Contingency Theory. Management Science 42, 187–204 (1996)
19. Clark Jr., T.D., Jones, M.C., Armstrong Curtis, P.: The Dynamic Structure of Management Support Systems: Theory Development, Research Focus, and Direction. MIS Quarterly 31, 579–615 (2007)
20. Dishman, P.L., Calof, J.L.: Competitive intelligence: a multispace precedent to marketing strategy. European Journal of Marketing 42, 766–785 (2007)
21. Choudhury, V., Sampler, J.L.: Information specificity and environmental scanning: An economic perspective. MIS Quarterly 21, 25–53 (1997)
22. Kajüter, P.: Die Regulierung des Risikomanagements im internationalen Vergleich. Zeitschrift für Controlling und Management 47, 12–25 (2004)
23. Sherman, W.S., Chambers, V.: SOX as Safeguard and Signal: The Impact of The Sarbanes-Oxley Act of 2002 on US Corporations’ Choice to List Abroad. The Multinational Business Review 17, 163–179 (2009)
24. Choo, C.W.: Information Use and Early Warning Effectiveness: Perspectives and Prospects. Journal of the American Society for Information Science and Technology 60, 1071–1082 (2009)
25. Webster, J., Watson, R.T.: Analyzing the Past to Prepare for the Future: Writing a Literature Review. MIS Quarterly 26, xiii-xxiii (2002)
26. IEEE: IEEE Standard Glossary of Software Engineering Terminology. IEEE Computer Society, New York (1990)
27. Gregor, S.: The Nature of Theory in IS. MIS Quarterly 30, 611–642 (2006)
28. Vom Brocke, J., Simons, A., Niehaves, B., Riemer, K., Plattfaut, R., Cleven, A.: Reconstructing the Giant: On the Importance of Rigour in Documenting the Literature Search Process. In: ECIS, Verona, pp. 2206–2217 (2009)

29. MIS: MIS Journal Ranking (AIS 2007), `http://ais.affiniscape.com/displaycommon.cfm?an=1&subarticlenbr=432` (accessed December 29, 2010)
30. Xu, X.M., Kaye, G.R., Duan, Y.: UK executives' vision on business environment for information scanning A cross industry study. Information & Management 40, 381–389 (2003)
31. Daft, R.L., Sormunen, J., Parks, D.: Chief Executive Scanning, Environmental Characteristics, and Company Performance: An Empirical Study. Strategic Management Journal 9, 123–139 (1988)
32. El Sawy, O.A.: Personal Information Systems for Strategic Scanning in Turbulent Environments: Can the CEO Go On-Line? MIS Quarterly 9, 53–60 (1985)
33. Cohen, J.B., Zinsbarg, E.D.: Investment Analysis and Portfolio Management. Dow Jones-Irwing Inc., Homewood (1967)
34. Taleb, N.N., Goldstein, D.G., Spitznagel, M.W.: The Six Mistakes Executives Make in Risk Management. Harvard Business Review 87, 78–81 (2009)
35. Gray, P.: From hindsight to foresight: Applying futures research techniques in information systems. Communications of the Association for Information Systems 22, 211–234 (2008)
36. Plambeck, N., Weber, K.: When the glass is half full and half empty: CEO's ambivalent interpretations of strategic issues. Strategic Management Journal 31, 698–710 (2010)
37. Mayer, J.H.: Organisatorische Veränderungen durch die aktuelle Wirtschaftskrise – Bestandsaufnahme und Implikationen für Unternehmenssteuerungssysteme. In: DCC, Dortmund, pp. 209–228 (2010)
38. Mayer, J.H., Wurl, H.-J.: Strategische Früherkennung in internationalen Konzernen, pp. 1–28. University of St.Gallen, St. Gallen (2011)
39. Porter, M.E.: Competitive Advantage: Creating and Sustaining Superior Performance. Free Press, New York (1985)
40. Frolick, M.N., Parzinger, M.J., Rainer Jr., R.K., Ramarapu, N.K.: Using EISs for environmental scanning. Information Systems Management 14, 35–40 (1997)
41. Salmeron, J.L.: EIS Evolution in Large Spanish Businesses. Information & Management 40, 31–40 (2002)
42. Dietz, J.L.G.: Architecture. Building strategy into design. Academic Service, The Hague (2007)
43. Arnott, D., Pervan, G.: Eight key issues for the decision support systems discipline. Decision Support Systems 44, 657–672 (2008)

Utilizing User-Group Characteristics to Improve Acceptance of Management Support Systems—State of the Art and Six Design Guidelines

Jörg H. Mayer[1], Robert Winter[1], and Thomas Mohr[2]

[1] University of St.Gallen, Institute of Information Management,
Mueller-Friedberg-Straße 8, 9000 St. Gallen, Switzerland
{joerg.mayer,robert.winter}@unisg.ch
[2] University of Darmstadt,
Hochschulstraße 1, 64289 Darmstadt, Germany
thomas.mohr@stud.tu-darmstadt.de

Abstract. In information systems (IS) design, understanding users and their preferences for interacting with IT is key. Such awareness is particularly important in the field of management support systems (MSS). We conduct a literature review on how user-group characteristics can be incorporated into MSS design and propose six design guidelines to enhance their adaptation capabilities. Three of these guidelines aim at better meeting users' functional preferences: incorporate more subjective information needs in MSS design, expand the scope of functional MSS principles beyond the user interface, and provide a comprehensive model of MSS functions for the growing number of expert users. Strengthen the constructional MSS perspective should make the design more concrete for practice. The fifth finding is to understand the characteristics of MSS usage by considering MSS use cases and access modes in addition to users' working style. Last but not least, MSS research should place more emphasis on principles for situational artefact design.

Keywords: Situational artefact design, working style, corporate business intelligence (BI), human-computer interaction (HCI).

1 Introduction

Both "management support systems" (MSS) [1] and "decision support systems" (DSS) [2] have been proposed as labels for information systems (IS) intended to provide IT support for managerial decision making. Since DSS evolved from a specific concept that originated as a complement to management information systems (MIS) and was overlapped in the late 1980s with executive information systems (EIS), we refer to our object of study as MSS instead [3]. This more general term was first used by Scott Morton [4] to cover DSS, EIS, knowledge management systems (KMS), and business intelligence (BI) [1].

Ideally, MSS design would meet the requirements of all potential users. But faced with limited resources, MSS design needs to balance standardization and individualization.

H. Jain, A.P. Sinha, and P. Vitharana (Eds.): DESRIST 2011, LNCS 6629, pp. 291–305, 2011.

In an analogy to situational method engineering [5], reference modeling [6], and design for artefact mutability [7] MSS designers might accomplish such a balance by segmenting requirements. Requirements are prerequisites, conditions, or capabilities needed by the users of a software systems [8]. The resulting design objective should then be to provide a situated solution for each segments by applying adaptation mechanisms for generic artefacts [9].

MSS design is often driven by manager's functional roles within the organization [10]. Such an IS design disregards their individual characteristics. Therefore, managers still question the relevance of MSS [11, 12]. In the light of these considerations, this article conducts a literature review on how user-group characteristics can be incorporated into MSS design. Based on the findings, we propose six design guidelines that enhance MSS adaptation capabilities to user-group preferences.

This article follows the tenants of design science research (DSR) in IS—an approach that focuses on developing innovative, generic solutions for important, relevant design problems in organizations and emphasizes utility [13]. Various processes exist for developing artefacts under the design science paradigm [14]. The one described by March and Smith [13], which distinguishes between "build" and "evaluate" activities, predominates in the literature [15]. Our overall research targets the "build" part and we follow vom Brocke et al.'s [16] five-step model with one modification: we arrive not only at future research questions [17], but design guidelines for MSS to provide a more concrete starting point for future research.

We motivate this article by reporting gaps in manager's MSS acceptance. After revisiting relevant foundations, we prove the relevance of user-group preferences for MSS design (Sec. 2) and derive a framework for categorizing the literature with findings from three IS fields (Sec. 3). We then describe the literature search process (Sec. 4.1). The most important publications surveyed provide accepted knowledge which we incorporate into our proposal (Sec. 4.2). Based on the findings, we state the research gaps in literature (Sec. 5) and develop design guidelines to overcome these gaps (Sec. 6). Three instantiations provide a first validation of our proposal. However, this article does not substantially evaluate the guidelines developed here or the subsequent design of MSS. These topics need to be addressed in future research (Sec. 7).

2 User Characteristics in MSS Research

As early as 1979, Zmud [18, p. 975] echoes several authors by claiming that "individual differences do exert a major force in determining MSS success." However, a few years later Huber's [19] stock-taking took the wind out of its sails for many years to come. He claimed that accommodating user preferences require IS designers to consider too many characteristics, that better educating users is a preferable solution, and that MSS might be completely configurable by users in future anyway.

The last 20 years invalidate Huber's line of argument. Research on user acceptance—the technology acceptance model (TAM) by Davis [20] and the IS success models by DeLone and McLean [21]—prove that user perception plays a predominant role in IS success. Moreover, the present moment seems especially favorable for redesigning MSS. Today's management grew up with gadgets of all kinds and have an increasingly positive attitude towards IT [22].

3 Structuring Requirements and Principles for MSS Design

Structuring our literature research, we first look at requirements engineering (RE, Sec. 3.1). Second, we concentrate on enterprise engineering (EE, Sec. 3.2) to move from requirements to principles of IS design. Third, we cover human-computer interaction (HCI, Sec. 3.3) to address user preferences and their IS interaction in detail. The findings are synthesized in a framework for our literature review (Sec. 3.4).

3.1 Requirements Engineering

RE is a field of software engineering concerned with determining the goals, functions, and constraints of hardware and software systems [23]. Herein, it is common to distinguish between functional and non-functional requirements [10, 24]. Functional requirements describe "what" the system should do. Thus, they are statements about a "function that a system … must be able to perform" [8, p. 35]. Related to user tasks within the organization, functional requirements are domain-specific. Customer relationship management systems or MSS are examples. Role-based access rights or drill-downs into products, regions, and customers are more detailed functional specifications of a MSS.

Non-functional requirements, in contrast, reflect "how well" the IS performs within the given environment as it fulfills its function [25]. They can be characterized as cross-domain. Examples are performance, usability, flexibility, reliability, or more in detail, IS response time [26]. Accurate customer relationship management systems or more business-driven MSS complete our example.

3.2 Enterprise Engineering

EE aims at the purposeful, theory-based design and implementation of enterprises from an engineering perspective [27]. Its central aspect is accomplishing the shift from users' requirements (black-box) to specification by design (white-box).

That may seem to closely resemble the objective of RE. Within EE, however, it is the result of the design phase [28]. That has two impacts: First, constructional requirements are exposed explicitly, introducing characteristics of the IS design process such as transparency, traceability and modularity. Second, the principles which result are predefined design actions specifying how MSS are brought to life, not just requirements [28-30]. A service-oriented IS architecture or a client-server architecture with thin clients are constructional principles not visible to the user, but important for the IS engineer and his IS design task.

3.3 Human-Computer Interaction

HCI is "… concerned with the ways humans interact with information, technologies, and tasks, ..." [31, p. 335]. Chan et al. [32] categorize three major themes. First, HCI concentrates on the users. Second, HCI is iterative, whereas the software development process validates that all design specifications are implemented. Third, the focus is on empirical testing, helping on IS design with the experience gained in practice.

The first step of HCI's user-centered design process is to understand the user characteristics and the IS context. Therefore, HCI has established several theories about human behavior and cognition that explain their responses to computer use [33] beyond RE (Sec. 3.1).

3.4 Framework for Literature Categorization

Following HCI research, we start for literature categorization with articles about techniques for user-group segmentation (A.1) and their characteristics (A.2). We go on with EE, which breaks up the IS design process into two stages. The black-box model ("usage" model) describes the user perspective and covers their functional requirements and associated functional principles (B.3). The white-box model covers the constraints to IS design from the engineering perspective. In addition to functional requirements and principles, IS design has to cover constructional requirements (C.1) and constructional principles (C.2) as well (Fig. 1).

Next, we detail the functional requirements by means of the RE findings. We go on with the separation of domain-specific requirements (B.1), which cover the purpose of IS design, and cross-domain requirements (B.2), which cover more formal aspects of IS as they fulfill their function. Finally, we add functional principles (B.3). Fig. 2 illustrates the framework.

4 Literature Analysis

After introducing our literature search strategy (Sec. 4.1), we synthesize the results on how user-group characteristics are incorporated into MSS design and in what areas are gaps to overcome (Sec. 4.2).

4.1 Search Strategy

Our literature search strategy follows vom Brocke et al.'s [16] four-step process: First, we focus on leading IS research outlets and select ten journals based on the catalog provided by the London School of Economics [34]. We consider this catalog as appropriate for our purposes, since it incorporates not only mainstream IS journals, but also social studies of IS.[1] Furthermore, we expand our list with proceedings from the two "A"-ranked international conferences listed by WKWI [35]: International and European Conferences on IS (ICIS, ECIS). To expand our sample towards engineering discipline, we look at outlets of systems and software engineering.[2] Finally, our search covers HCI journals[3].

---

[1] We choose five journals of each set, namely: MIS Quarterly, Information Systems Research, Information & Management, Journal of Management Information Systems, Decision Support Systems as well as European Journal of Information Systems, Information & Organization, Information Systems Journal, Journal of Organizational and End-User Computing, and Journal of Information Technology.

[2] We used several journal rankings [35-37] and choose Information and Software Technology, Communication of the ACM, ACM Computing Surveys, Journal of Systems and Software, and the International Journal of Systems Science as the ones with a great impact factor.

[3] We found Human-Computer Interaction, International Journal of Human-Computer Interaction, International Journal of Human-Computer Studies, and Computers in Human Behavior in the journal rankings [35-37] and added AIS Transaction on Human-Computer Interaction as an upcoming HCI Journal.

To access the journals, we choose EBSCOhost, Science Direct and ProQuest, as they predominantly cover issues of the last 20 years. Then, we execute the keyword search on titles and abstracts[4]. The results were 466 hits, of which we found 20 to be relevant. To identify further articles, we do a backward search which leads to a total of 30 relevant articles.

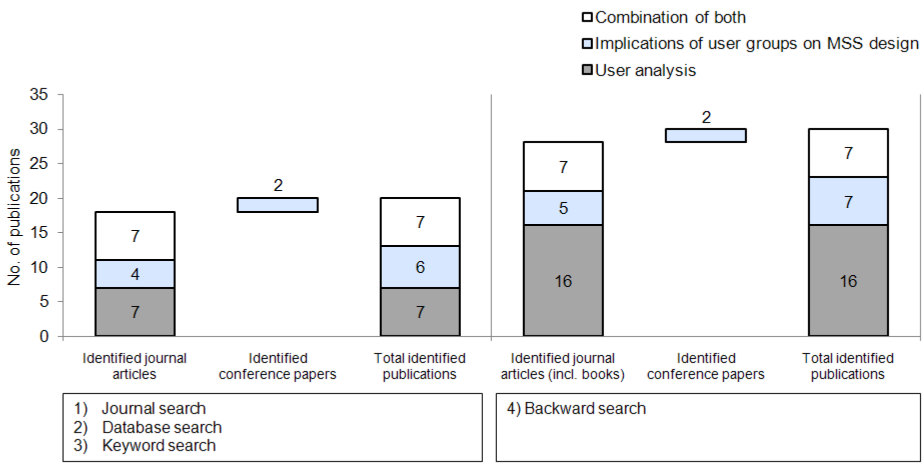


Fig. 1. Selection of relevant publications

4.2 Results

Structured according to the framework presented in Sec. 3, Figure 2 exhibits identified related work. Studies relating to more than one component of the framework appear more than once. Hereafter, the publications from the journals with the highest impact factors[5] are highlighted and described briefly to get a "look & feel" for the following seven clusters of publications.

A User Analysis

A.1 User-Group Segmentation. These articles are rooted in psychology and deal with an individual's cognitive style. That is the way in which individuals tend to grasp information (e.g. quantitatively vs. qualitatively) and how they apply this information when making decisions (e.g. logical argumentation vs. intuition). We limit our scope to those that have been applied in IS research.

---

[4] Search String: "decision making" OR "executive information system" OR "decision support system" OR "management information system" OR "data warehouse" OR "business intelligence") AND ("use" OR "style" OR "pattern" OR "adoption" OR "acceptance."

[5] We consider impact factors from http://www.elsevier.com.

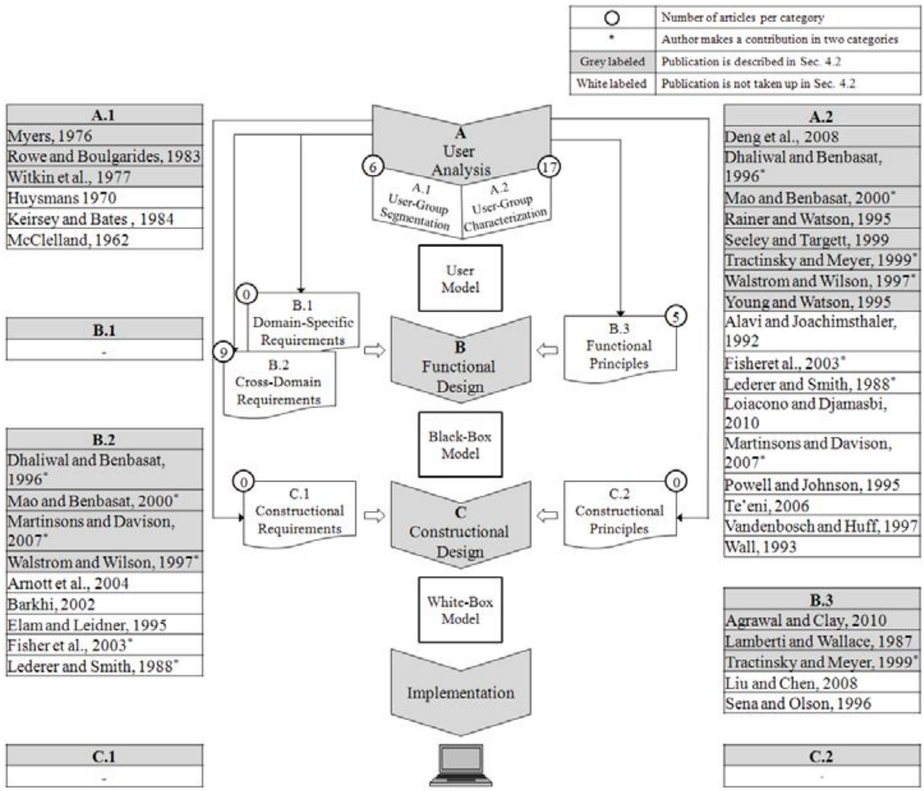


Fig. 2. Classification of researched publications

One of the most popular and widespread technique for user-group segmentation is the Myers-Briggs Type Indicator (MBTI) [38]. This assessment classifies an individual's personality according to four dichotomies: attitude, perceiving function, judging function, and lifestyle. With respect to decision support and decision making, the perceiving and judging functions are of particular interest.

Related to the judging functions of MBTI is Witkin's concept of field-dependence and field-independence [39]. Field-dependent individuals perceive data in their context as a whole and are less attentive to detail (low analytical). Field-independent people perceive data independent of their context, paying more attention to details (high analytical).

A two-dimensional classification of cognitive decision styles is provided by Rowe and Boulgarides [40]. According to the individual's values and their cognitive complexity between tolerance ambiguity and need for structure, four styles are differentiated: Individuals with an analytic preference enjoy solving problems and use careful analysis. Conceptual preferences are achievement-oriented and initiate new ideas (need for recognition). Directive preferences expect results and follow their intuition (need for power). Individuals with a behavioral preference are supportive and use limited data (need for affiliation).

Summarizing the findings, there are a lot of methods to differentiate individual's cognitive style (Figure 2), which can be considered to separate user groups for MSS design. We found no research gap in this concern.

A.2 User-Group Characterization. The first set of articles here covers differentiating characteristics that have an impact on MSS. Dhaliwal and Benbasat [41] and Mao and Benbasat [42] demonstrate that the users' level of expertise has an effect on their MSS usage. Deng et al. [43] identify cultural differences in MSS user satisfaction. Tractinsky and Meyer [44] claim that an MSS user is not only in a receiving, but also a presenting role. Consequently they demonstrate that a users' objective such as facilitating decision making or maintaining an impression has an impact on his or her preferred interface design.

Each article in the second group develops IS profiles for a certain user group. They are defined as a set of directly observable user characteristics. Young and Watson [45], and Rainer and Watson [46] argue that executives are not typical MSS users. They demonstrate that "ease of use" has no significant effect on executives' acceptance of MSS and derive success factors such as previous computer experience. Walstrom and Wilson [47] go on to derive three user types of executives: converts, pacesetters, and analyzers. The first group uses MSS to improve information access; the second uses MSS to improve communication and performance monitoring; and the third uses MSS to solve problems. Seeley and Targett [48] identify four patterns of executive computer use over time: steady-state users, growing users, born-again users, and declining users. Summarizing the findings, there is even more literature about user-group characterization and their IT usage than on user-group segmentation (A.1) so that we cannot state a research gap in this concern.

B Implications of User Groups for Functional MSS Design

B.1 Domain-Specific Requirements. MSS literature provides several methods for determining information needs and corresponding reference models for its specification. As a starting point, Rockart's [49] Critical Success Factors (CSF) focus on single executives and interviews. An example of a recently developed model is the Balanced Scorecard (BSC) [50].

In terms of the model perspective, Palvia et al. [51] focus on external information to be included in the MSS: competitors, markets, economics, finance etc. Internal information is specified by Mayer and Marx [52]. With financial and management accounting, compliance and program management, as well as cash flow and liquidity management they propose five information clusters for their new-generation EIS. In summary, this kind of literature tends to go more and more beyond the financials. But none of these proposals considers interlinkages between domain-specific requirements, user preferences and MSS design. A reason could be that the theory of homo oeconomicus dominates management research in the last decades [53] and thus to some extent IS research on its linking edge between human being and IT as well.

B.2 Cross-Domain Requirements. Martinsons and Davis [54] draw implications for the predominant decision style in America, Japan, and China. Chinese leaders will consider using MSS only to support routine decision making or informal personal reporting.

That differs from that of American business leaders for quantitative reasoning even in complex situations. Therefore, they need MSS that codify valuable information. Japanese business leaders also tend to apply systematic online reporting.

Dhaliwal and Benbasat [41] examine the different explanation needs of novices and experts from MSS. They find out that novices tend to make greater use of "feed-forward" explanations (non-case-specific, generalized information on the input cues for decision making), while experts make greater use of "cognitive feedback" (case-specific information that explains the outcome of an analysis). Regarding explanations of the "cognitive feedback" type, Mao and Benbasat [42] show that novices need "reasoning-trace" explanations ("how") more than "justification" ("why").

Walstrom and Wilson [47] define typical functionalities to be used by converts, pacesetters, and analyzers. Converts use MSS to access predefined reports, analyzers will use the MSS primarily to perform analysis, whereas pacesetters make extensive use of communication capabilities. Research on implication of user groups on cross-domain requirements has already been done. However, they are often not compatible among each other due to different user group characterizations.

B.3 Functional Principles. The following articles develop concrete principles for MSS design. Tractinsky and Meyer [44] derive functional principles depending on the objectives of MSS users from a presenting perspective. If the reports in the MSS are used to aid decision making, the interface should be restricted to 2D bars and figures so as to not distract from the content. If the reports in the MSS are used for presentation, the interface should apply 3D bars and figures.

Lamberti and Wallace [55] design user interfaces for portraying uncertainty in an MSS for the military—identifying critical targets in a real-time environment. Based on the dimensions of field-dependence/field-independence and systematic/heuristic decision style, they develop four screen setups with different colors, symbols, graphics, and numeric displays.

Agrawal and Clay [56] examine the effect of an individual's temperament, categorized by the Keirsey Temperament Sorter [57], and information representation on decision making. For tabular representations, individuals with a guardian or artisan temperament will have higher decision accuracy. For graphical representations, individuals with an idealist or rational temperament will have higher decision accuracy. For tabulars, individuals with a guardian temperament will have lower decision time than those with an artisan temperament. For graphical representations, individuals with an idealist temperament will have lower decision time than those with rational temperament. In comparison to the number of articles regarding the functional requirements (Sec. B.1, B.2), there is no facetted body of knowledge available.

C Implications of User Group for Constructional MSS Design

In the field of *constructional requirements* (C.1), Walia and Carver [58] identify and classify errors that occur during the requirement phase and develop a taxonomy of these. Therein, they list people errors, documentation errors as well as process errors, which include, for instance, management errors or traceability errors. EE also deals with constructional requirements such as the reduction of system complexity and costs [27, 28].

In the field of *constructional principles* (C.2), it is primarily the software engineering discipline that deals with IS architecture styles on a high level, such as a service-oriented architecture [10]. Although some authors even deal with architecture styles, none of the articles considers implications of user-group characteristics on constructional requirements or constructional principles.

5 Synthesis

Six findings emerge from the literature analysis. They will determine the guidelines for enhancing the MSS capabilities to adapt to user-group preferences (Sec. 6) and are as follows.

First finding: An imbalance is evident between studies regarding the impact of user-group preferences on domain-specific and on cross-domain (functional) requirements. While a number of publications exist on user-group preferences and their impact on cross-domain requirements (B.2), we found none examining their impact on domain-specific requirements (B.1). This absence suggests that subjective information needs is not a big issue in MSS literature so far.

Second finding: There is a lack of functional principles addressing issues beyond user-interface design. Although cross-domain requirements (B.2) are extensively described in the literature, design principles (B.3) can only be found rarely, and the ones we found are primarily limited to the user interface [42, 55].

Third finding: A commonly accepted model for a basic set of MSS functions is not available. Functional requirements (B.1) tend to take a granular focus on specific MSS functions, such as generating explanations or providing information on data quality, but they are often not aligned within a commonly accepted overall (research) model for MSS functions such as Walstrom and Wilson [47]. Such a model would be beneficial for creating a common body of knowledge and integrating more and more user-centered functions into MSS design in future.

Fourth finding: No publications exist on the effect of user-group preferences on constructional requirements and principles. As Figure 2 shows, most of the publications address the impact of user-group preferences on the black-box model (B.1-B.3). We found no publications that cover implications of user-group preferences on the white-box model (C.1, C.2). One reason could be that white-box elements focus on concrete MSS design and thus are more difficult to evaluate.

Fifth finding: Determining MSS usage characteristics based on working style alone is no longer sufficient. The literature we found on user-group preferences focuses on cognitive style, users' IS perception and their decision making by different user types (A.1). Only a few contributions consider other MSS user factors such as gender, level of expertise, and seniority (A.2). More importantly, in addition to MSS users' working style, other user factors, such as MSS use cases or MSS access modes, are barely considered at all.

Sixth finding: Concrete principles for leveraging state-of-the-art user-group characteristics are underrepresented compared to the associated requirements. The last finding is that studies of user-group preferences and their impact on MSS requirements (B.1, B.2, and C.1) dominate over those on MSS principles (B.3 and C.2) per se. This imbalance could be due to the fact that such principles are broader in scope and therefore more difficult to evaluate. However, this research gap leads to incoherent solutions for individual problems instead of generic artefacts with capabilities allowing adaptation for classes of design problems [9].

6 Evaluation

Two instantiations at large, international companies in chemicals (Europe; employees: 105,000; sales: USD 65 bn) and automotive supply (Europe; 150,000; 32) as well as one in an international consultancy (U.S., 7,000; n/a) helped us to evaluate the findings in practice and formulate guidelines for the required MSS adaptation capabilities. In the first two instantiations, our findings provided a basis for designing an EIS user interface in line with executives' preferences. The third one was about a tool to track time and expenses. We performed the evaluation by interviews with the executives as the IS user and the IS engineers and compared their state of the art IS with the findings from the instantiation.

First guideline: Differentiate functional requirements and incorporate more subjective information needs in MSS design. The first finding of our synthesis outlined the imbalance between the rich body of knowledge on user preferences impacting cross-domain requirements (B.2) and the lack of literature on domain-specific requirements (B.1). In terms of what domain-specific MSS adaptation capabilities are required, our instantiations showed that prevailing opinion views objective information needs, determined by an executive's tasks within the organization, as more important than subjective ones [59]. At the same time, providing an adaptation mechanism to achieve a situational balance for meeting these two types of information needs would help to improve user acceptance of MSS.

Second guideline: Expand the scope of functional MSS principles to areas beyond user-interface design. Such functional principles can take the form of standardized dialogs or drill-down functionalities to provide a consistent platform for distinct user groups. From our instantiation we see a preference such as net sales analysis by product, region, and most important customers—sequentially and in that order. Another example is a cash flow graph over time or most important KPIs such as EBIT (earnings before interest and taxes) in comparison to other KPIs.

Third guideline: Develop a comprehensive model of basic MSS functions even for expert users. In our instantiations we recognized that the number of analytical power users on the C level has grown significantly [60]. MSS design should incorporate appropriate MSS principles, such as basic reporting and planning functions, more experienced shortcut navigation or flexible analyses such as ad hoc reporting, non-routine information, and direct links to upstream IS.

Fourth guideline: Place greater emphasis on examining the effect of user-group preferences on the constructional MSS perspective. Our research revealed a lack of information about user-group characteristics on the white-box MSS model. Our instantiations showed that a design process prospectively involving the user is important for IS acceptance. In this way, engineers understand the concrete implications of user-group characteristics on constructional requirements and principles, helping to ensure that adaptation capabilities are not hampered by, i. e. an inappropriate architecture, and—vice versa—business requirements with no grounding in available MSS capabilities should be prevented right from the start of MSS design.

Fifth guideline: Consider in MSS design not only working style, but use case and access mode as well to determine an executive's usage. In addition to the working style, MSS use case (e.g., individual analysis, one-to-few working meetings, or one-to-many presentations) and MSS access mode (e.g., stationary, portable, or mobile) are important user factors in MSS design.The consulting case provides an example: When they rolled out the new tool to track time and expenses, they failed to consider the user-group preferences of its consultants. Since confidentiality prevents them from working for their clients during travel, trips provide them with the opportunity to catch up with administrative tasks. Unfortunately, the new tool required an Internet connection, which limited its accessibility during travel. If this requirement had been considered in its development, the tool might have been more successful. Our instantiations showed that a combination in terms of MSS users' working style, use case, and access mode will lead to MSS design that is more appropriate even for idiosyncratic managers.

Sixth guideline: Emphasize concrete principles for (situational) artefact design per se rather than "pure" identification of requirements. Our instantiation showed that requirements analysis alone will not contribute to better MSS design. Thus, principles should play a larger role in achieving better MSS design and a requirements analysis must be complemented with concrete functional and constructional principles to avoid incoherent solutions for individual problems in favor of generic solutions with adaptation capabilities for different classes of design problems.

7 Outlook and Future Research

The objective of this article was to review the literature on how consideration of user-group characteristics can be incorporated into MSS design. We proposed six design guidelines that enhance the capabilities of MSS to adapt to user preferences. Three of these guidelines involve looking at user characteristics to improve functional MSS design. Our synthesis also reveals a need to strengthen the constructional perspective. Our fifth guideline calls for supplementing information on MSS users' working style with MSS use case and MSS access mode as to create a complete view of MSS usage. Last, but not least, research should emphasize concrete principles rather than "pure" requirements identification to force coherent solutions—in other words, generic artefacts with mechanisms for adapting them for different classes of design problems. Three instantiations provided a first validation of our proposal.

The research on hand is limited to a restricted number of researched publications. However, the fact that we covered the leading journals means that major contributions should be included. An extension of our work will expand our analysis, but this should be a secondary issue. Focusing on DSR, it is more important to specify the design guidelines presented here with future "build" and "evaluate" activities—driven in additional case studies or in a survey to gain a direct perspective on user-group preferences for MSS in a relevant sample.

Our future research will be twofold. On the one hand, we will improve the literature analysis as outlined. On the other, we will propose a MSS user-group characteristics taxonomy based on a validated segmentation of working styles, use cases, and MSS access modes. Such a taxonomy will then allow to develop situational MSS design methods.

References

1. Clark Jr., T.D., Jones, M.C., Armstrong Curtis, P.: The Dynamic Structure of Management Support Systems: Theory Development, Research Focus, and Direction. MIS Quarterly 31, 579–615 (2007)
2. Arnott, D., Pervan, G.: Eight Key Issues for the Decision Support Systems Discipline. Decision Support Systems 44, 657–672 (2008)
3. Power, D.J.: Decision Support Systems: A Historical Overview. In: Burstein, F., Holsapple, C.W. (eds.) Handbook on Decision Support Systems 1: Basic Themes, pp. 121–140. Springer, Berlin (2008)
4. Scott Morton, M.S.: Computer-Driven Visual Display Devices - Their impact on the Mana-gement Decision-Making Process. Ph. D. dissertation, Harvard Business School (1967)
5. Bucher, T., Klesse, M., Kurpjuweit, S., Winter, R.: Situational Method Engineering - On the Differentiation of “Context” and “Project Type”. In: Situational Method Engineering - Fundamentals and Experiences, pp. 33–48. Geneva Springer, Berlin (2007)
6. Becker, J., Delfmann, P., Knackstedt, R.: Adaptive Reference Modeling: Integrating Configurative and Generic Adaptation Techniques for Information Models. In: Becker, J., Delfmann, P. (eds.) Reference Modeling, pp. 27–58. Physica, Heidelberg (2007)
7. Gregor, S., Jones, D.: The Anatomy of a Design Theory. Journal of the Association for Information Systems 8, 312–335 (2007)
8. IEEE: IEEE Standard Glossary of Software Engineering Terminology. IEEE Computer Society, New York (1990)
9. Winter, R.: Problem Analysis for Situational Artefact Construction in Information Systems. In: A Tribute to Marco de Marco (2010)
10. Sommerville, I.: Software Engineering. Pearson, München (2010)
11. Pettey, C., Stevens, H.: Gartner EXP Worldwide Survey of Nearly 1,600 CIOs Shows IT Budgets in 2010 to be at 2005 Levels, `http://www.gartner.com/it/page.jsp?id=1283413` (accessed April 16, 2010)
12. Wixom, B.H., Watson, H.J.: The BI–Based Organization. International Journal of Business Intelligence Research 1, 13–28 (2010)
13. March, S.T., Smith, G.F.: Design and Natural Science Research on Information Technology. Decision Support Systems 15, 251–266 (1995)
14. Peffers, K., Tuunanen, T., Gengler, C.E., Rossi, M., Hui, W., Virtanen, V., Bragge, J.: The Design Science Research Process: A Model for Producing and Presenting Information Systems Research. In: 1st International Conference on Design Science in Information Systems and Technology, Claremont, pp. 83–106 (2006)

15. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28, 75–105 (2004)
16. vom Brocke, J., Simons, A., Niehaves, B., Riemer, K., Plattfaut, R., Cleven, A.: Recon-structing the Giant: On the Importance of Rigour in Documenting the Literature Search Process. In: 17th European Conference on Information Systems, Verona, pp. 2206–2217 (2009)
17. Webster, J., Watson, R.T.: Analyzing the Past to prepare for the Future: Writing a Literature Review. MIS Quarterly 26, 13–23 (2002)
18. Zmud, R.W.: Individual Differences and MIS Success: A Review of the Empirical Literature. Management Science 25, 966–977 (1979)
19. Huber, G.P.: Cognitive Style as a Basis for MIS and DSS design: Much Ado about Nothing? Management Science 29, 567–579 (1983)
20. Davis, F.D.: Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. MIS Quartely 13, 318–340 (1989)
21. DeLone, W.H., McLean, E.R.: The DeLone and McLean Model of Information Systems Success: A Ten-Year Update. Journal of Management Information Systems 19, 9–30 (2003)
22. Pijpers, G.G.M., Bemelmans, T.M.A., Heemstra, F.J., van Montfort, K.A.G.M.: Senior Executives' Use of Information Technology. Information and Software Technology 43, 959–971 (2001)
23. Laplante, P.A.: Requirements Engineering for Software and Systems. CRC Press, Boca Raton (2009)
24. IFPUG: Function Point Counting Practices Manual - Release 4.3.1. International Function Point User Group (IFPUG), Princetion Junction (2010)
25. Robertson, S., Robertson, J.: Mastering Requirements Engineering. Addison-Wesley, Boston (2006)
26. Paech, B., Kerkow, D.: Non-Functional Requirements Engineering - Quality is Essential. In: Requirements Engineering: Foundation for Software Quality, Riga REFSQ, Essen, pp. 27–40 (2004)
27. Hoogervorst, J.A.P.: Enterprise Governance and Enterprise Engineering. Springer, Berlin (2009)
28. Dietz, J.L.G.: Architecture. Building Strategy into Design. Academic Service, The Hague (2007)
29. IEEE: IEEE Recommended Practice for Architectural Description of Software Intensive Systems (IEEE Std 1471-2000). IEEE Computer Society, New York (2000)
30. Stelzer, D.: Enterprise Architecture Principles: Literature Review and Research Directions. In: Workshop on Trends in Enterprise Architecture Research, Stockholm, pp. 21–35 (2009)
31. Zhang, P., Benbasat, I., Carey, J., Davis, F., Galletta, D.: Human-Computer Interaction Research in the MIS Discipline. Communications of the Association for Information Systems 9, 334–355 (2002)
32. Chan, S.S., Wolfe, R.J., Fang, X.: Issues and Strategies for Integrating HCI in Masters Level MIS and E-Commerce Programs. International Journal of Human-Computer Studies 29, 497–520 (2003)
33. Lyytinen, K.: HCI Research: Future Directions that Matter. AIS Transactions on Human-Computer Interaction 2, 22–26 (2010)
34. Willcocks, L.P., Whitley, E.A., Avgerou, C.: The Ranking of Top IS Journals: A Perspective from the London School of Economics. European Journal of Information Systems 17, 163–168 (2008)
35. Die Sprecher der Wissenschaftlichen Kommission Wirtschaftsinformatik im Verband der Hochschullehrer für Betriebswirtschaft (WKWI) und des Fachbereichs

Wirtschaftsinformatik der Gesellschaft für Informatik (GI-FB WI): WI-Journalliste 2008 sowie WI-Liste der Konferenzen. Wirtschaftsinformatik 50, 155–163 (2008)
36. AIS: MIS Journal Rankings, http://ais.affiniscape.com/displaycommon.cfm?an=1&subarticlenbr=432 (accessed November 20, 2010)
37. VHB: Teilranking Wirtschaftsinformatik und Informationsmanagement, http://vhbonline.org/service/jourqual/jq2/teilranking-wirtschaftsinformatik-und-informationsmanagement/ (accessed November 25, 2010)
38. Myers, I.B.: Introduction to Type. Center for Applications of Psychological Type, Gainesville (1976)
39. Witkin, H.A., Moore, C.A., Goodenough, D.R., Cox, P.W.: Field-Dependent and Field-Independent Cognitive Styles and Their Educational Implications. Review of Educational Research 47, 1–64 (1977)
40. Rowe, A.J., Boulgarides, J.D.: Decision Styles - A Perspective. Leadership & Organizational Development Journal 4, 3–9 (1983)
41. Dhaliwal, J.S., Benbasat, I.: The Use and Effects of Knowledge-based System Explanations: Theoretical Founcdations and a Framework for Empirical Evaluation. Information Systems Research 7, 343–362 (1996)
42. Mao, J.-Y., Benbasat, I.: The Use of Explanations in Knowledge-Based Systems: Cognitive Perspectives and a Process-Tracing Analysis. Journal of Management Information Systems 17, 153–179 (2000)
43. Deng, X., Doll, W.J., Al-Gahtani, S.S., Larsen, T.J., Pearson, J.M., Raghunathan, T.S.: A Cross-Cultural Analysis of the End-User Computing Satisfaction Instrument: A Multi-Group Invariance Analysis. Information & Management 45, 211–220 (2008)
44. Tractinsky, N., Meyer, J.: Chartjunk or Goldgraph? Effects of Presentation Objectives and Content Desirability on Information Presentation. MIS Quarterly 23, 397–420 (1999)
45. Young, D., Watson, H.J.: Determinates of EIS Acceptance. Information & Management 29, 153–164 (1995)
46. Rainer, R.K., Watson, H.J.: What Does It Take for Successful Executive Information Systems? Decision Support Systems 14, 147–156 (1995)
47. Walstrom, K.A., Wilson, R.L.: An Examination of Information Systems (EIS) Users. Information & Management 32, 75–83 (1997)
48. Seeley, M., Targett, D.: Patterns of Senior Executives' Personal Use of Computers. Information & Management 35, 315–330 (1999)
49. Rockart, J.F.: The Changing Role of the Information Systems Executive - A Critical Success Factors Perspective. Sloan Management Review 24, 3–13 (1982)
50. Kaplan, R.S., Norton, D.P.: The Balanced Scorecard: Translating Strategy into Action. Harvard Business School Press, Boston (1996)
51. Palvia, P., Kumar, A., Kumar, N., Hendon, R.: Information Requirements of a Global EIS: An Exploratory Macro Assessment. Decision Support Systems 16, 169–179 (1996)
52. Mayer, J.H., Marx, F.: Managing a Paradox – Design Principles for Executives' IT Support. In: 14th Pacific Asia Conference on Information Systems, Taipeh, pp. 1321–1332 (2010)
53. Brousseau, É., Glachant, J.M.: New Institutional Economics – A Guidebook. Cambridge University Press, Cambridge (2008)
54. Martinsons, M.G., Davison, R.M.: Strategic Decision Making and Support Systems: Comparing American, Japanese and Chinese Management. Decision Support Systems 43, 284–300 (2007)

55. Lamberti, D., Wallace, W.A.: Presenting Uncertainty in Expert Systems: An Issue in Information Portrayal. Information & Management 13, 159–169 (1987)
56. Agrawal, A., Clay, P.: Temperament and Cognitive Fit: An Empirical Investigation of Task Performance. In: 31st International Conference on Information Systems, Saint Louis, pp. 1–10 (2010)
57. Keirsey, D., Bates, M.: Please Understand Me: Character and Temperament Types. Gnosology Books (1984)
58. Walia, G.S., Carver, J.C.: A Systematic Literature Review to Identify and Classify Software Requirement Errors. Information and Software Technology 51, 1087–1109 (2009)
59. Nutt, P.C.: Evaluating MIS Design Principles. MIS Quarterly 10, 139–158 (1986)
60. Mayer, J.H., Stock, D.: Nutzertypen für die situative FIS-Gestaltung: Ergebnisse einer empirischen Untersuchung. In: 10th International Conference on Wirtschaftsinformatik WI 2011, Zurich, pp. 139–149 (2011)
61. Huysmans, J.H.B.M.: The Effectiveness of the Cognitive-Style Constraint in Implementing Operations Research Proposals. Management Science 17, 92–104 (1970)
62. McClelland, D.C.: Business Drive and National Achievement. Harvard Business Review 40, 113–122 (1962)
63. Alavi, M., Joachimsthaler, E.A.: Revisiting DSS Implementation Research: A Meta-Analysis of the Literature and Suggestions for Researchers. MIS Quarterly 16, 95–116 (1992)
64. Fisher, C.W., Chengalur-Smith, I., Ballou, D.P.: The Impact of Experience and Time on the Use of Data Quality Information in Decision Making. Information System Research 14, 170–188 (2003)
65. Lederer, A.L., Smith, G.L.: Individual Differences and Decision-Making Using Various Levels of Aggregation of Information. Journal of Management Information Systems 5, 53–69 (1988)
66. Loiacono, E., Djamasbi, S.: Moods and Their Relevance to Systems Usage Models within Organizations: An Extended Framework. AIS Transactions on Human-Computer Interaction 2, 55–73 (2010)
67. Powell, P.L., Johnson, J.E.V.: Gender and DSS Design: The Research Implications. Decision Support Systems 14, 27–58 (1995)
68. Te'eni, D.: Designs that Fit. In: Zhang, P., Galletta, D. (eds.) Human-Computer Interaction and Management Information Systems: Foundations, pp. 205–224. M.E. Sharpe, London (2006)
69. Vandenbosch, B., Huff, S.L.: Searching and Scanning: How Executives Obtain Information from Executive Information Systems. MIS Quarterly 21, 81–107 (1997)
70. Wall, K.D.: A Model of Decision Making under Bounded Rationality. Journal of Economic Behavior and Organization 21, 331–352 (1993)
71. Arnott, D., Jirachiefpattana, W., O'Donnell, P.: Executive Information Systems Development in an Emerging Economy. Decision Support Systems 42, 2078–2084 (2007)
72. Barkhi, R.: Cognitive Style Mitigate the Impact of Communication Mode. Information & Management 39, 677–688 (2002)
73. Elam, J.J., Leidner, D.G.: EIS Adoption, Use, Impact: The Executive Perspective. Decision Support Systems 14, 89–103 (1995)
74. Liu, Y., Chen, A.N.K.: The Effect of Individual Differences, Tasks, and Decision Models on User Acceptance of Decision Support Systems. In: 14th Americas Conference on Information Systems, Toronto, pp. 1–11 (2008)
75. Sena, J.A., Olson, D.H.: Decision Support for the Administrative Man: A Prototype DSS Case. European Journal of Information Systems 5, 10–23 (1996)

How Service Orientation Can Improve the Flexibility of Executive Information Systems—An Architecture Reworked from a Business Perspective

Jörg H. Mayer

University of St.Gallen, Institute of Information Management,
Mueller-Friedberg-Straße 8, 9000 St. Gallen, Switzerland
joerg.mayer@unisg.ch

Abstract. In recent years, service orientation has been discussed as a new design paradigm promising better manageability and changeability of increasingly complex IS. This article examines their role in executive information systems (EIS) design and contributes a reworked architecture that is more flexible than the state of the art. Structured in terms of four layers strategy, organization, alignment, and IT support it uses cross-layer modeling chains that enable even drill-through analyses when needed. Mapping loosely coupled services within an alignment layer provides the necessary flexibility. Two typical changes in financial accounting and management accounting processes at a telecom company provided an opportunity to evaluate the reworked architecture. Finally, the lessons learned helped us in two ways: providing concrete starting points for integrating service orientation into EIS architecture design and arguing for the reworked architecture on hand.

Keywords: Enterprise architecture (EA), informations systems (IS) integration, corporate business intelligence (BI), pilot implementation.

1 Introduction

Companies today operate in an increasingly dynamic environment. Due to their overall responsibility, C-level managers, hereafter referred to as executives, are particularly affected by this situation. Information systems (IS) meant to help these top managers are known as executive information systems (EIS). They are designed to serve as their central, hands-on, day-to-day source of information [1].

EIS have become complex over recent years, and their IT side is often unable to keep pace with the changes in business [2]. Thus, their flexibility[1] gains importance [5]. This need became more apparent as the 2008/2009 economic crisis catalyzed two changes in corporate management [6]: executives have expanded their role in day-to-day business and they have to make decisions faster than they have in the past.

[1] Flexibility is the cost-adequate, qualitatively good capability to handle new but anticipated requirements [3]. Agility covers responses to unexpected requirements as well [4]. Hereafter, we use flexibility to cover both abilities.

H. Jain, A.P. Sinha, and P. Vitharana (Eds.): DESRIST 2011, LNCS 6629, pp. 306–320, 2011.

Directly (point-to-point) linking business and IT artefacts often fails to meet flexibility needs. To provide flexible links, researchers often propose employing service orientation in IS design, but this term has become a buzzword in recent years. Concrete projects are rare in practice, often due to a lack of support from the business. In the wake of the 2008/2009 economic crisis, however, businesses should now more clearly see the need for IS flexibility, and even executives should begin to recognize its value. At the same time, they likely find approaches such as virtual decoupling [7] difficult to understand, and we predict that they see little applicability for them in practice [8].

Thus, this article examines the role of service orientation in EIS design not from the perspective of an IS engineer, but from the business perspective of executives dealing with current changes in corporate management. Based on the findings, we contribute a reworked architecture that is more flexible than the state of the art.

This article adheres to design science research in IS [9]. Distinguishing constructs, models, methods, and instantiations [10], we set out a model of an EIS architecture. The guidelines for integrating service orientation help to specify how the model should be designed to increase artefact flexibility [11].

Various processes exist for developing artefacts under the design science paradigm [9, 10, 12]. Emphasizing "build" and "evaluate" activities, we apply the research process of Peffers et al. [12]. The motivation comes from current changes in corporate management and their impact on EIS design. After establishing the foundation and presenting the results of a literature review (Sec. 2), we clarify gaps in the body of knowledge with the results of a survey (Sec. 3). Four design layers derived from enterprise architecture (EA) research provide a framework for structuring both the empirical requirements analysis and the reworked EIS design (Sec. 4). We then discuss how the reworked EIS architecture is applied (Sec. 5.1-5.2). The lessons learned (Sec. 5.3) help us to provide concrete starting points for integrating service orientation into EIS architecture design and to argue for the reworked architecture presented here (Sec. 6). The article ends with a summary and topics for future research (Sec. 7).

2 Need for a Service-Oriented Rework of EIS Architecture

We focus on service orientation as a design paradigm for IS [13] and consider *IS* as the entirety of persons, business processes, and IT processing data and information in an organization [14]. Hence an IS includes both business and IT artefacts. Service orientation involves composing complex solutions from a set of loosely coupled building blocks. Herein, s*ervices* are functional capabilities—known in software development as encapsulated functionalities—with published interfaces [15]. The greater flexibility that results from service orientation makes it possible to realign business and IT artefacts when the business changes or technological innovations occur. This capability is based on the two properties that define the concept. First, service consumers do not need to know which actions are being performed to produce a result (transparency), and second, services produce a meaningful, distinct result for a given context (granularity). Both properties foster the loose coupling that allow an IS consisting of services to flexibly change.

According to ANSI/IEEE 1471 [16] and ISO/IEC 42010 [17], *architectures* are defined as (a) "[t]he fundamental organization of a system, embodied in its components, their relationships to each other and to the environment," and as (b) "the principles governing its design and evolution." The organization of a system is often represented by models. A *model* represents an extract of reality by selecting objects relevant for expressing a certain domain and/or purpose. Architecture models can serve a variety of purposes. Lyytinen [18] distinguishes three types: those that describe an individual IS; environmental models, which describe interactions between IS; and IS context models, the type we employ here.

Existing Architecture Models: Several IS context models have been developed [19]. Those currently accepted as state of the art are the Architecture of Integrated Information Systems (ARIS) framework [20], the Zachmann Framework [21], and The Open Group Architectural Framework (TOGAF) [22], one of the most frequently used EA in Europe. All these models provide taxonomies and relations for expressing the basic elements of EA. ARIS includes four dimensions: organization, function, data, and processes. The Zachman Framework and TOGAF are more layered frameworks. The first is structured in terms of scope, enterprise and system model, technology, and detailed representations. The latter employs the layers business, application, data, and technology architecture. Comparing these approaches, Aier et al. [23] distilled four design layers for proper IS architectures: strategy, organization, alignment, and IT support (software and IT infrastructure). We use this structure as the starting point for both our survey (Sec. 3) and the rework of the EIS architecture to address the requirements indicated by participants (Sec. 4).

EIS-specific Architecture Design: No such clear picture exists regarding EIS architectures. Following Walls et al. [11], EIS design is based on requirements and subsequent design recommendations in the form of reference models. For example, Rockart [24] employed a method of providing information to top management called "Critical Success Factors." An example of a more recent model is the Balanced Scorecard [25]), but a comprehensive approach that combines financial and management accounting with compliance management and project management is lacking.

Koutsoukis et al. [26] argue that data warehouse (DWH) technology offers good availability and performance, allowing integration of data from various transactional and analytical sources. Using online analytical processing for data models provides flexibility. Mayer and Krönke [27] argue for ad hoc reporting capabilities and direct links to upstream systems. Internal information is specified by Mayer and Marx [28]. With financial and management accounting, compliance and program management, as well as cash flow and liquidity management they propose five information clusters for their new-generation EIS. Chen [29] proposes an integrated organization and information model. All in all, however, none of the EIS architectures examined stress the flexibility to respond to changing corporate management processes. Furthermore, constructional principles, such as TOGAF, are not applicable for all design layers of an IS architecture and they are not specified for the EIS domain.

EIS Architecture Design and Service Orientation: Within EIS architecture design, the role of the service orientation is unclear [30]. Yang and Xin [31] propose using it and thus shifting EIS scope from data models to the flexible (re-)use of functions, but they give no guidelines for doing so. In conclusion, no publications were found that highlight service orientation in EIS design.

It seems that a fully integrated IS approach would be the way to address the issues mentioned above. But a look back reveals that "total approaches" failed. By the early 1980s, comprehensive management information systems (MIS) overtaxed mainframes and integration capabilities, and we believe that this would still happen with "modern" IT. We therefore look for a trade-off between the all-encompassing MIS approach and "purely" strategic—but in terms of IT architecture integration—isolated EIS.

3 Survey

Specifying our proposed business perspective on EIS architecture design (Sec. 1), we select a *field survey* (cross-section analysis) as the research method with the following characteristics. Since corporate management has most clearly become impossible without IT in large, international companies, this type of organization was defined as the population for our survey. A paper-based questionnaire was sent to the CEOs and CFOs of the 250 largest companies listed in the Financial Times "Europe 500" report. The first part of the survey took place between April and June 2008, shortly before the onset of the economic crisis in mid-2008. The follow-up survey was conducted between November 2009 and March 2010 to identify a perspective on "lessons learned" from the crisis. A study among companies listed in the DAX® 100 [32] in 1999 provided the reference base. As a result, the research is classified as a longitudinal survey over the last 11 years on executives' work and their IS support.

A total of 59 CEOs and CFOs responded in the 2008 survey (59/500: 11.8%). In the 2010 survey, 42 questionnaires were returned (42/500: 8.4%). Of these respondents, 30 executives returned both questionnaires (50.8%/71.4%).

Following the "organizational imperative" [33], EIS design must consider two context factors: they must be integrated into the company's organization and oriented toward executives' corporate management activities. Starting with the organizational context factors, we examined two changes due to 2008/2009 economic crisis.

More Operational Responsibility, even at Headquarters: According to our 2008 survey results, one-third of the executives said that they currently intervene frequently and extensively in operations—in parallel to their strategic management tasks. Another 22% characterized their involvement as very frequent and very extensive. These findings show that executives expanded their role in operations due to the 2008/2009 economic crisis. The 2010 results indicate that this trend started 2008 in the financial sector, is now evident in the industrial sector as well: again, one-third of the executives said their involvement in operations is frequent and extensive, while another 10% specify they intervene very frequently and very extensively.

Faster Decision Making and Biased EIS Objectives: A second question examined executives' demand for flexibility. Almost 50% answered that they are operating in a environment that is more aggressive than ever. A situation that continues in the 2010 survey. As a consequence, executives have to make decisions faster than they have in the past. In terms of the two objectives for EIS architecture design, increasing flexibility and lowering costs [34], executives rate the first as "high" to "very high." While executives often viewed IS just as a "cost pool" in the past, the survey shows that they now consider flexibility to be of similar importance as cost (Fig. 1).

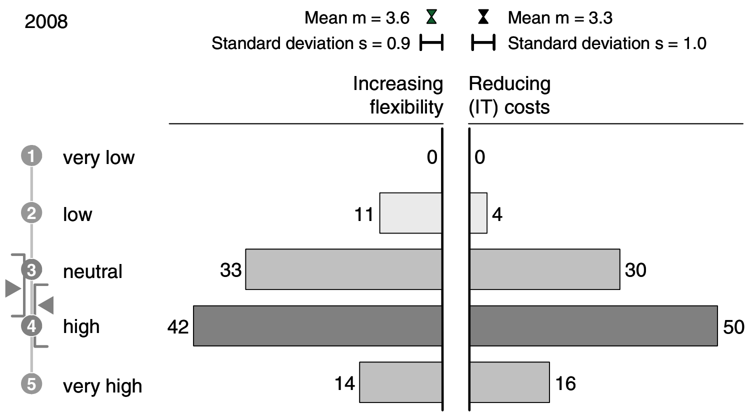


Fig. 1. Executives' perspective on current EIS architecture objectives

With a third question we looked at the areas in which executives forsee the greatest need for change. In addition to changes in organizational structures (mean of 3.1) and the data models (3.6), they outline the need to respond flexibly to changing management processes (4.0). Thus, defining the objective of EIS design more in detail, we asked the executives about their main activities (strategy layer, see Sec. 2) and then about their most important corporate management processes (organization layer) and how they allocate their day-to-day time to these activities (Table 1).

Table 1. Tasks of executives (EIS strategy and organization layer)

Category	Task	1999 [%]	2008 [%]	2010 [%]	Δ 1999/2008 [%]	Δ 2008/2010 [%]
Strategy layer						
Ratio of external communication vs. internal management	External communication	11	36	30	+25	-6
	Internal management	89	64	70	-25	+6
- Internal management in detail	Normative management	13	17	17	+4	+/-0
	Strategic leadership	55	48	46	-7	-2
	Group services	14	10	14	-4	+4
	Cash flow and liquidity management	12	16	19	+4	+3
	Others	6	9	4	+3	-5
- External communi-cation in detail	Focused on regulatory compliance	n/a	2	2	n/a	+/-0
	Biased toward regulatory compliance	n/a	0	3	n/a	+3
	Balance between regula-tory compliance/capital market communication	n/a	59	55	n/a	-4
	Biased toward capital market communication	n/a	38	35	n/a	-3
	Focused on capital market communication	n/a	1	5	n/a	+4
Organization layer						
Strategy management	Strategy definition	42	32	37	-10	+5
	Strategy execution and strategy tracking	58	68	63	+10	-5

4 EIS Architecture Design with Four Layers: Strategy, Organization, Alignment, and IT Support

The following section describes how the identified requirements have been implemented within a reworked EIS architecture to answer the first research question (Sec. 1). In addition to the constructional principles, architectures cover the "blueprint" of the artefact to be designed, thus, specifying their components and the links among them. Accordingly, both artefacts and an associated principle for each design layer are proposed below. The results can be summarized as follows (Fig. 2).

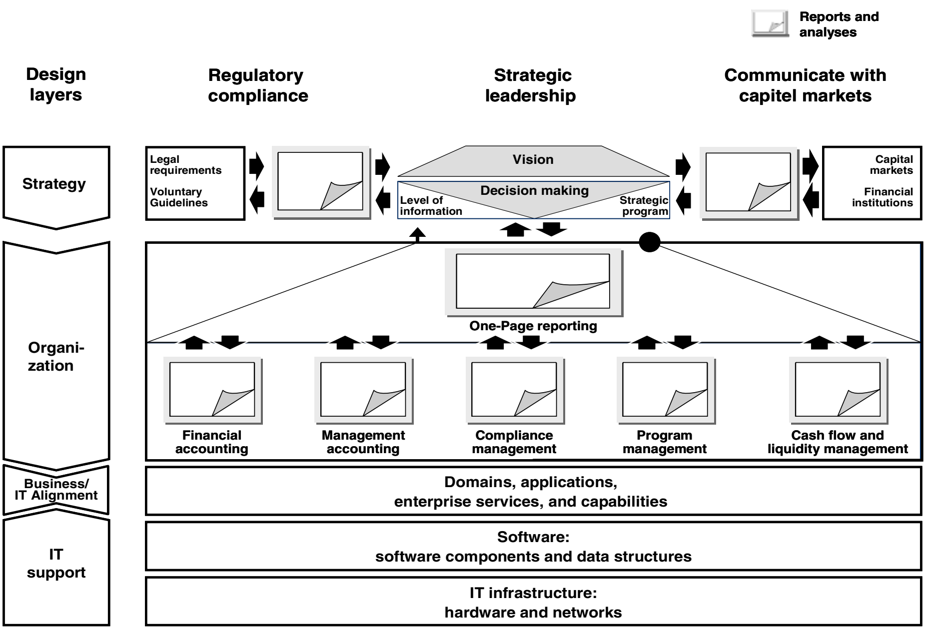


Fig. 2. Proposal for a service oriented EIS architecture with four design layers

4.1 Strategy

Executives devote the majority of their day-to-day time to internal management and, within this area of activity, to strategic leadership (Table 1). As a result, we propose as the objective of a reworked EIS architecture to support the latter (Fig. 2), while bearing in mind that executives are intervening more frequently and extensively in operations and thus need more operational information (Sec. 3). As the role of external communication has approximately tripled in the last 11 years, from 11% (1999) to 36% (2008) and 30% (2010, Table 1), supporting such efforts is defined as a second objective of a reworked EIS. Finally, regulatory compliance is essential (Table 1). Thus, unlike its single-purpose EIS predecessors, a reworked EIS architecture must accomplish a threefold goal: *supporting strategic leadership and communicating the results to capital markets while ensuring regulatory compliance*.

4.2 Organization

Defining strategy may be the more intellectually demanding task, but the survey results show executives devote considerably more time to executing and tracking strategy than to strategy definition (Table 1). We propose a two-step structure for reporting on these efforts to "translate strategy into action" and—if deviations from planning occur—making decisions to correct the course. Using the body of knowledge (Sec. 2, [28]), a one-page report should provide an overview and serve as a starting point for detailed predefined analysis, while a flexible periphery should allow ad hoc queries, access to nonroutine information, and direct links to upstream systems.

In keeping with the specification "breadth before depth" [35], we propose that reports cover the most important information clusters only [28]: financial and management accounting, compliance and program management, and cash-flow and liquidity management. The design principle can be summarized as follows: *a two-step reporting approach should close the strategy execution and tracking loop with one-page reporting and detailed analyses, both covering five information clusters*.

4.3 IT Support: Software and IT Infrastructure

IT support for the proposed EIS architecture has two aspects (Sec. 2): software and IT infrastructure. **Software**: Since the 21% and 23% of the executives considered using standard software in EIS architecture to be important or very important, we follow TOGAF Principle 5, "Common Use Applications" [22], by taking a "buy before build" approach—in other words, using standard business software packages. In terms of data structures—conceptual, logical, and physical data models—we follow TOGAF's Principle 10, "Data is an Asset," and propose a group DWH to centrally maintain all EIS data. The resulting design principle for the EIS software support layer is thus: *to ensure flexibility at acceptable cost, standard software components should be combined individually and data structures should be configured unambiguously*.

IT infrastructure: Our framework entails no specific requirements regarding hardware and networks. First implementations of this approach [27, 35] show that the standard software components and data structures used create no special demands on IT infrastructure. *Ensuring scalable, stable, and resource-efficient hardware and networks* is an appropriate directive for the IT infrastructure.

4.4 Business/IT Alignment

Because increasing flexibility and lowering (IT) costs matter in the "new" normal IS design equally to executives, we recommend unbundling business and IT artefacts following the service orientation paradigma in IS. This decoupling is supported by domains, (logical) applications, and capabilities, and by enterprise services.

In the accounting field, services represent activities such as "perform payment" or currency conversion. Bundles of IT functionalities that are not structured according to services but rather the information objects they manage or a particular corporate management process are labeled as capabilities. Applications cover tasks such as accounts payable, or accounts receivable in the financial accounting domain, value-driver trees or margin calculations in the management accounting domain. Business domains are characterized by a large number of connected applications and services. Thus, financial or management accounting are such examples. *Supporting flexible links between corporate management processes, the relevant software components, and data structures while lowering IT costs* can summarize the principle of the business/ IT alignment layer.

5 Implementing the Reworked Architecture

Our instantiation shows results from one of the world's leading telecom companies, which offers its customers the entire spectrum of IT and telecommunications services from a single source. In 2009, they generated revenue of EUR 64.6 billion with about 260,000 employees. We proposed our approach as a more flexible solution than the state of the art, as both the business side and IT department recognized that they could contribute better to executives' IS support if they could more flexibly handle changes within the corporate management process. In the following, we discuss service orientation for financial reporting (Sec. 5.1), drill-through analyses (Sec. 5.2), and lessons learned (Sec. 5.3).

5.1 Legal Group Consolidation Process

The obligation to compile an annual statement is regulated by the generally accepted accounting principles for the region in question. For our telecom company, these rules are documented in § 242 I of the German Commercial Code (Handelsgesetzbuch). As a prerequisite for the group perspective, the results of the legal consolidation process—such as group net sales or earnings before income and taxes—must be incorporated into the annual statement (§ 290 Handelsgesetzbuch).

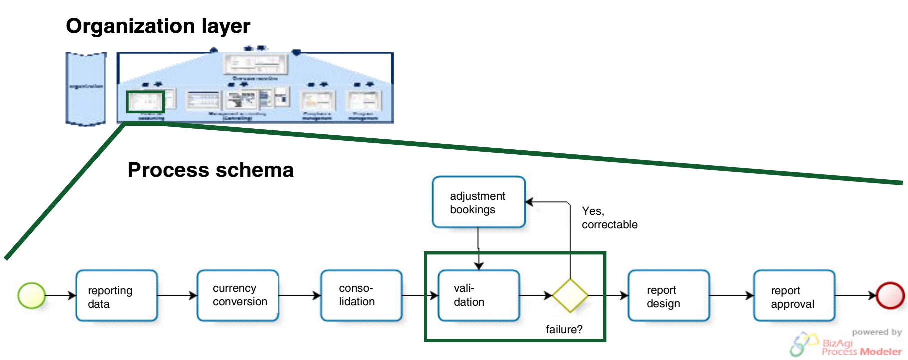


Fig. 3. Legal group consolidation process in the organization layer

Modeling at the organization layer: Based on legal requirements (strategy layer), a new, more service-oriented group consolidation process must be designed at the organizational layer. To do so, appropriate service patterns are derived from the consolidation process. Business Process Modeling Notation (BPMN), a graphical notation for specifying business processes in a process model, was used for documentation [37]. The first step in the process is to capture the divisional statements ("reporting data"). The data are converted into the group currency and consolidated. Then, the results are validated. Adjustment bookings are performed if any failures occur. The final step is to design and approve the report (Fig. 3).

Translation to the alignment layer: Applications serve to refine this process. It is necessary to specify whether each activity is carried out automatically or manually. Unless a failure occurs, automated activities involve no interaction with human users and can be performed separately within IS. Currency conversion of divisional statements is one such service (Fig. 4, layer 2). For manual activities, authorities must be provided with the information they need to make decisions. For example, the "adjustment bookings" activity (Fig. 4, layer 1) is detailed in the alignment layer with a string of enterprise services, such as "failure report (automatic), require adjustment booking (automatic), and check adjustment booking."

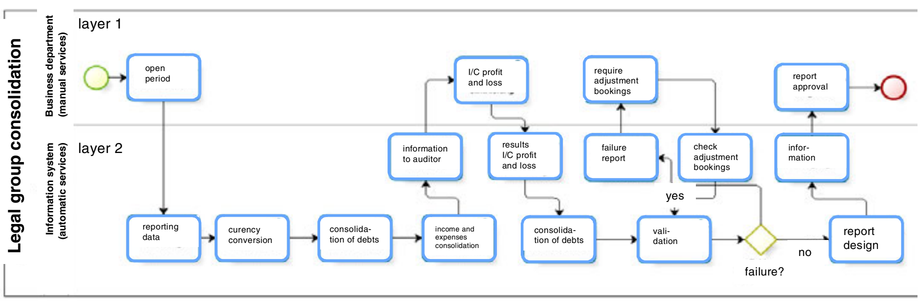


Fig. 4. Legal group consolidation in the business/IT alignment layer

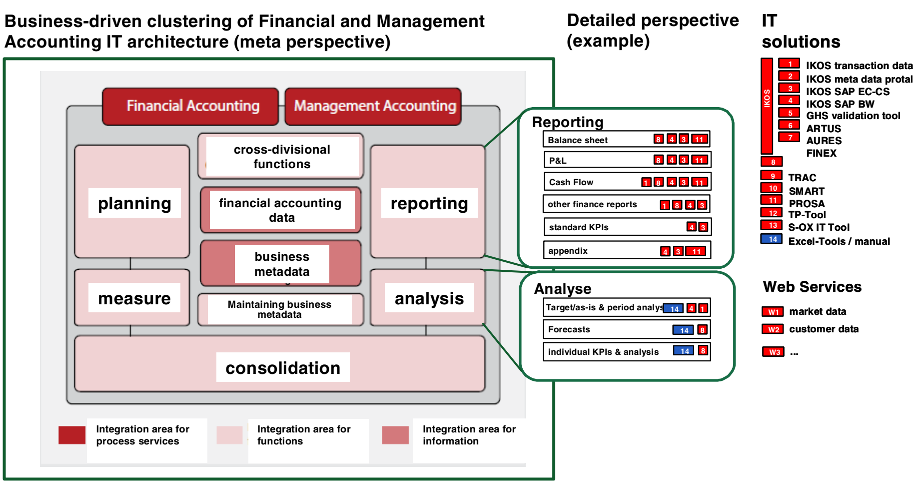


Fig. 5. Domain map covering the business perspective on IT architecture

Translation to the software layer: A last step of the service-oriented legal consolidation process is to link the services of the alignment layer with existing IT functionalities. Creating these links involves taking a business perspective on the IT architecture (Fig. 5). The IT functionalities were identified using open interfaces of SAP Business Consolidation (BCS) and SAP Financials (FI), both standard software. In the service-oriented approach described here, web services are used because they allow technology- and location-independent referencing.

5.2 Generating "Drill-Through" Analysis

The discussion of the strategy layer pointed out the increasing significance of operational information for executives (Sec. 2). In response, a reworked EIS architecture should provide reporting that drills through to the underlying upstream systems. A first step in our instantiation was to implement direct access to the financial reporting module of the enterprise resource planning system (ERP). If the EIS provides condensed versions of the balance sheet and P&L and cash-flow statements, the drill-throughs allow direct access to the full versions, such as the complete list of accounts receivable and payable. In our architecture, the procedure for generating such a *flash report* is based on the consolidation process described previously.

Legal consolidation (activity 3, Fig. 3) requires a great deal of time. From an internal perspective, however, just a "simple summation" would be sufficient for a first view. In the reworked EIS architecture, it is easy to add a new activity to the legal consolidation process to perform this step.

Changes at the organization layer: The group-currency statements from the divisions on group currency must be captured and then a summarized P&L must be compiled and presented for approval (Fig. 6). To avoid delaying the consolidation processes, reporting data should be duplicated for this purpose.

Changes at the alignment and software layers: In our instantiation, two new services—"duplicate reporting data" and "create flash report"—are added to align the process flow. Finally, a link was created with SAP BCS to allow the new function "create flash report" to be performed automatically. The DWH perform the "duplicate reporting data" function.

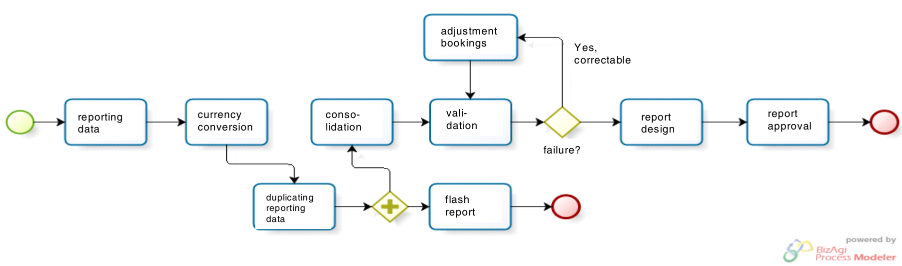


Fig. 6. Service oriented design of flash report in conjunction with legal group consolidation

The reworked architecture also makes it possible to create value-driver trees or calculate different kinds of contribution margins. Drill-through into operational production and sales figures show the determinants of important products. The program management view (Fig. 2), in turn, offers details of strategic initiatives. For example, executives can access the working status of new production lines or sales distribution activities in terms of budget, schedule, and responsibilities. Finally, executives are able to switch from the legal structure (group and divisions) to one based on the management structure (group, divisions, and key business units). Within this view, they can look at net sales by region or key figures, such as margins, for the most important customers.

5.3 Five Lessons Learned towards Service Orientation in EIS Architectures Design from a Business Perspective

Regarding the second research question of how service orientation contributes to the flexibility of EIS architecture, our survey and the instantiation presented here provide five insights that can help drive service orientation to rework EIS architecture design.

First lesson learned: Executives currently see flexibility as the greatest challenge for EIS architecture design. EIS architecture design must become more flexible for two reasons: first, to handle changes in the corporate management process, and second, to provide the additional information executives want from upstream ERP IS via "drill-throughs." Service orientation can provide this flexibility by cross-layering modeling chains with services at an alignment layer (Fig. 2).

Second lesson learned: Despite the desire for flexibility, service orientation has not yet been implemented due to a current focus on short-term payoff. The 2008/2009 economic crisis brought greater cost transparency, and the payoff time for investments became shorter than ever. But implementing service orientation and reaping the flexibility benefits is a long-term project. In this longer view, service orientation should be able to help while providing a positive side effect: reusing services, such as planning and consolidation, by identifying them in more than one IS will save costs. In our instantiation, looking at IS from a business perspective (Fig. 5) made it possible to identify "currency conversion" in four financial software solutions. This activity was centralized as an enterprise service and maintenance was leveraged for a single functionality. As a result, cost and time were reduced by 30% for this IS accounting function.

Third lesson learned: Measuring the business benefits of service orientation is a strong lever for its future success. Our instantiation showed that service orientation must demonstrate its impact in terms of business benefit. These benefits can be measured in dimensions such as cost efficiency or consistent ease of handling through different IS. Furthermore, a business-driven domain model of IS (Fig. 5) help the business better understand what capabilities are available. Integrating service orientation into EIS architecture design does not have to result in the "best" solution, but the most efficient in economic terms.

Fourth lesson learned: A top-down approach offers the best service-oriented EIS architecture model. As the scope of executive tasks expands, architectural efforts should aim to more strongly reflect a company's vision and strategic program and help to translate these targets into action (strategy and organizational layer). Our instantiation showed that reports and analyses can be used to align business requirements with IT capabilities. Thus executives must take an active role in the process, not only to clarify their information needs, but for the first prototype reviews as well. The latter is an area in which they are often unfamiliar, so they are best served by providing them a hands-on "look and feel" prototype and then enhancing it step by step. Following this design approach, efforts to promote service orientation should be a top-down, business-driven project and should be communicated that way.

Fifth lesson learned: Further domains exist for service orientation. Using service orientation should be feasible in domains beyond EIS. Business metadata on intercompany ownership relations or addresses, customer data about discounts, or other order information seem to be such areas. These data are needed by several IS within the company and must be consistent to allow uniform group standards and a single face to the customer. A good approach would be to outsource these data to a separate IS, maintain them centrally, and provide them as an enterprise services from this "single point of truth" to different company authorities and IS.

6 Evaluation

In terms of the reworked EIS architecture to better respond to changes in corporate management, some methodical points should be discussed regarding how this work compares with the state of the art and what future issues remain to be addressed.

The architecture model presented here was proposed to focus on the business perspective on EIS design. Thus, the requirements have been broadly validated by a *survey* applying three characteristics (Sec. 3). First, this survey was not just a one-time effort, but a longitudinal survey over the last 11 years. Second, we reached out to the biggest companies listed in the FT "Europe 500" report, assuming that they have the resources to let highly experienced experts handle these issues. Third, we asked the executives directly about their requirements for IS support and service orientation incorporated in their EIS architecture design in detail.

EIS are often implemented as "islands" devoted solely to issues of strategic leadership, but today they should integrate different management methods within a consistent business/IT architecture. Thus, our EIS architecture design combines strategic and executive frameworks with IS architecture models. For the first, we researched a number of management methodologies EIS should support, such as Critical Success Factors and Balanced Scorecards. Understanding what executives do and their perspective on EIS design allows a focus on pressing current issues for the business side, especially supplying executives with operational information via "drill-throughs". In terms of the IS architectures models, we researched ARIS and TOGAF. On this point, the survey imparted a clearer perspective on current requirements across architecture layers than looking at the state of the art alone. Finally, structuring the survey and the design in terms of a multilayer model provided a reliable business-to-IT starting point.

The instantiations so far made it possible to generate a first perspective on our architecture's flexibility, but user requirements for such EIS vary more than we expected; e.g., needs differ widely between dashboard-oriented CFOs and CIOs and HR directors who work still with paper-based standard reports. A modular architecture design like the one presented here, with adaptation mechanisms and distinct information clusters, should help IS engineers to handle change better than the state of the art does. To reflect the arguments that have gone before, we refer to our findings as a reworked architecture for EIS (Fig. 7).

1st generation EIS (early 1990ies)	Reworked architecture
Business requirements not aligned with IT capabilities	*Requirements validated from a business perspective across companies listed in FT "Europe 500" report*
Strategic leadership islands, no integration with operational IS	*Consistent business-to-IT architecture*
Individual solutions; missing adaptation mechanisms	*Modular design with adaptation mechanism for different classes of design issues*

Fig. 7. Characteristics of the reworked EIS architecture

7 Conclusion and Future Research

At first glance, the design of EIS architectures seems to be established. However, our current survey within companies listed in the FT "Europe 500" report identified two basic changes in corporate management which impacts executives' IS support. Their range of implementation deviates enough from existing approaches to justify referring to the EIS architecture presented here as a "reworked" approach.

Starting with a literature review, we identified how corporate management activities shifted in the 2008/2009 economic crisis and examine their impact on EIS architecture design. Our survey specified the requirements from a business perspective. The reworked EIS architecture meets these requirements with a four layer design; strategy, organization, alignment, and IT support. An instantiation helped us to demonstrate and evaluate flexibility, highlighting the incorporated service orientation.

Some limitations to this study should also be noted. First, our literature research is based on a restricted number of publications. However, the fact that we covered leading journals means major contributions should be included. In terms of the survey, only internal EIS stakeholders participated, providing 59 and 42 data sets. However, the samples of other executive surveys are no larger [38, 39]. Thus, expanding the literature review is worthwhile, but should be a secondary issue.

In terms of design science research, it is important to continue with "build" and "evaluate" activities to improve the proposed EIS architecture (Sec. 4) and its service orientation (Sec. 5). Our future research will use more case studies to determine the generalizability of the proposed model and, hopefully, forthcoming extensions.

Beyond the IS architecture design, we expect further innovations. A younger management generation, more familiar with IT, will be particularly interested. New user interfaces and end-user devices should simplify IS handling, even for executives.

We are currently investigating this issue with a follow-up study to identify executive working profiles in order to develop a more situational approach to EIS design.

References

1. Walters, B.A., Jiang, J.J., Klein, G.: Strategic information and strategic decision making: the EIS/CEO interface in smaller manufacturing companies. Information & Management 40, 487–495 (2003)
2. Clark Jr., T.D., Jones, M.C., Armstrong, C.P.: The Dynamic Structure Of Management Support Systems: Theory Development, Research, Focus, And Direction. MISQ Executive 31, 579–615 (2007)
3. Schelp, J., Winter, R.: Integration Management for Heterogeneous Information Systems. In: Desouza, K.C. (ed.) Agile Information Systems—Conceptualizations, Construction, and Management, Oxford (2007)
4. Yusuf, Y.Y., Sarhadi, M., Gunasekaran, A.: Agile manufacturing: the drivers, concepts and attributes. International Journal of Production Economics 62, 33–43 (1999)
5. Luftman, J.N.: Key Issues for IT Executives 2004. MIS Quarterly Executive 29, 269–285 (2005)
6. Mayer, J.H.: Organisatorische Veränderungen durch die aktuelle Wirtschaftskrise – Bestandsaufnahme und Implikationen für Unternehmenssteuerungssysteme. Tagungsband des 25. Deutschen Controlling Congress, Reichmann, T (Hrsg.), Neue Anforderungen an das Management & Controlling, Dortmund, 209–228 (2010)
7. Aier, S., Winter, R.: Virtual Decoupling for IT/Business Alignment – Conceptual Foundations, Architecture Design and Implementation Example. Business & Information Systems Engineering (BISE) 1, 150–163 (2009)
8. Friedrich-Alexander-University Erlangen-Nürnberg/Detecon: Die geschäftsfokussierte Informationstechnologie—Business-IT-Alignment als zentrales Steuerungsinstrument zur strategischen Ausrichtung der IT, Erlangen-Nürnberg und Bonn (2009)
9. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28, 75–105 (2004)
10. March, S.T., Smith, G.F.: Design and Natural Science Research on Information Technology. Decision Support Systems 15, 251–266 (1995)
11. Walls, J.G., Widmeyer, G.R., El Sawy, O.A.: Building an Information System Design Theory for Vigilant EIS. Information Systems Research 3, 36–59 (1992)
12. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems 24, 45–77 (2007)
13. Aier, S., Winter, R.: Critical Success Factors of Service Orientation in Information Systems Engineering - Derivation and Empirical Evaluation of a Causal Model. Research paper. University of St.Gallen, St.Gallen (2010)
14. Vessey, I., Ramesh, V., Glass, R.L.: Research in Information Systems: An Empirical Study of Diversity in the Discipline and Its Journals. Journal of Management Information Systems 19, 129–174 (2002)
15. Dietsch, A., Goetz, T.: Nutzenorientiertes Management einer Serviceorientierten Unternehmensarchitektur. Wirtschaftsinformatik 47, 10–21 (2005)
16. IEEE 1471. The Institute of Electrical and Electronics Engineers, Inc. – Recommended Practice for Architectural Description of Software-Intensive Systems. IEEE Computer Society, New York (2000)

17. ISO/IEC 42010. International Organization for Standardization/International Electrotechnical Commission - Systems and software engineering – Recommended Practice for Architectural Description of Software-Intensive Systems (2007)
18. Lyytinen, K.: Different Perspectives on Information Systems: Problems and Solutions. ACM Computing Surveys 19, 5–46 (1987)
19. Schekkerman, J.: How to survive in the jungle of enterprise architecture frameworks: Creating or choosing an enterprise architecture framework, 2nd edn., Trafford (2006)
20. Scheer, A.W.: Wirtschaftsinformatik: Referenzmodelle für industrielle Geschäftsprozesse. 7. Aufl., Berlin (1997)
21. Zachman, J.A.: A Framework for Information Systems Architecture. IBM Systems Journal 26, 277–293 (1987)
22. The Open Group: The Open Group Architectural Framework Version 9, `http://www.opengroup.org/togaf` (assessed March 2, 2011)
23. Aier, S., Riege, C., Winter, R.: Unternehmensarchitektur – Literaturüberblick und Stand der Praxis. Wirtschaftsinformatik 50, 292–304 (2008)
24. Rockart, J.F.: Chief executives define their own data needs. Harvard Business Review 57, 81–93 (1997)
25. Kaplan, R.S., Norton, D.P.: Balanced Scorecard: Translating Strategy into Action. Harvard Business School Press, Boston (1996)
26. Koutsoukis, N.-S., Mitra, G., Lucas, C.: Adapting on-line analytical processing for decision modelling: interaction of information and decision technologies. Decision Support Systems 26, 1–30 (1999)
27. Mayer, J.H., Krönke, B.: Financial Reporting bei der Continental AG. ERP 6, 44–46 (2010)
28. Mayer, J.H., Marx, F.: Managing a Paradox – Design Principles for Executives' IT Support. In: 14th Pacific Asia Conference on Information Systems, Taipeh, pp. 1321–1332 (2010)
29. Chen, M.: A model-driven approach to accessing managerial information: the development of a repository-based executive information system. Journal of MIS 11, 33–63 (1995)
30. Wu, L., Barash, G., Bartolini, C.: A service-oriented architecture for business intelligence. In: IEEE International Conference on Service-Oriented Computing and Applications, Newport Beach (2007)
31. Yang, X., Xin, Z.: A Study on the Integration Model of EIS Based on SOA, Bejing, pp. 627–633 (2008)
32. Wurl, H.J., Mayer, J.H.: Ansätze zur Gestaltung effizienter Führungsinformationssysteme für die internationale Management-Holding - Ergebnisse einer empirischen Untersuchung. Zeitschrift für erfolgsorientierte Unternehmenssteuerung (Controlling) 11, 13–21 (1999)
33. Markus, M.L., Robey, D.: Information Technology and Organizational Change: Causal Structure in Theory and Research. Management Science 34, 583–598 (1998)
34. Wilkinson, M.: Designing an "adaptive" enterprise architecture. BT Journal 24, 81–92 (2006)
35. Aier, S., Kurpjuweit, S., Saat, J., Winter, R.: Enterprise Architecture Design as an Engineering Discipline. AIS Transaction on Enterprise Systems 1, 36–43 (2009)
36. Mayer, J.H., Winter, R.: Corporate Navigator—A New-generation Executive Information System: Lessons from BASF, research paper. University of St.Gallen, St.Gallen (2010)
37. OMG: Business Process Modeling Notation (BPMN) Specification (2006), `http://www.bpmn.org/Documents/OMG%20Final%20Adopted%20BPMN%201-0%20Spec%2006-02-01.pdf` (accessed December 15, 2010)
38. Seeley, M., Targett, D.: Patterns of senior executives' personal use of computers. Information & Management 35, 315–330 (1999)
39. Walstrom, K.A., Wilson, R.L.: An examination of information systems (EIS) users. Information & Management 32, 75–83 (1997)

How IT-Based Co-creation Can Provide Small Banks Access to the Financial Market – A Prototype Development from a Design Science Research Perspective

Sven Weber and Roman Beck

E-Finance Lab, Goethe University, Grüneburgplatz 1,
60323 Frankfurt, Germany
{svweber,rbeck}@wiwi.uni-frankfurt.de

Abstract. The asset-backed securities (ABS) market shrank due to the financial crises in 2007/2008. The financial collapse was caused by both the consumer and the commercial ABS markets failure. However, the ABS market is an important source of funding for market participants in the financial services industry. Hence, we see a returning trend for ABS markets and present in this paper the development of a prototype from a design science research (DSR) perspective to enable smaller banks to securitize and sell their assets. Given the relatively low mortgage and credit volumes within smaller banks they are currently not able to package and sell their securities. The prototype developed in this paper offers a solution by pooling all mortgages gathered from hundreds of smaller banks, calculating the attached risk, and finally, selling them to investors. In this context, DSR enabled us to build such a cooperative strategy in form of a prototype.

Keywords: Asset-backed Securities, Design Science Research Approach, Small Banks, Financial Services Sector.

1 Introduction

Securitization of financial assets such as mortgages in the financial services industry has enabled banks to create sub-prime financial instruments. These instruments contributed to the financial crises, e.g., through asset-backed securities (ABS) which are bonds backed by the cash flow of a variety of pooled receivables or loans [2]. However, creating an ABS was only possible for larger financial institutions with a large amount of assets (e.g. mortgages) to trade. In addition, these institutions needed the knowhow and compute power to securitize them. Smaller players such as cooperative or savings banks have not the compute power, amount of mortgages, as well as knowhow to securitize and trade their assets individually. Therefore, they have a lesser competitiveness compared to larger banks since they have to comply to the same financial regulations such as the Basle II accord nevertheless. Given the importance of smaller banks for national financial systems, we develop in this paper a securitization prototype that provides securitization-as-a-service for savings and loan banks. The prototype was developed in a three year research project together with

H. Jain, A.P. Sinha, and P. Vitharana (Eds.): DESRIST 2011, LNCS 6629, pp. 321–334, 2011.

partners from industry following a design science research (DSR) approach [21, 29, 39]. The DSR approach allowed us to specify, implement, and evaluate the prototype to provide new insights for the domain under consideration. The developed ABS prototype selects potential mortgages for securitization from a database where all smaller banks store their mortgages at a central IT service provider. Subsequently, the prototype automatically provides the decision makers at each bank the opportunity to sell or not to sell the mortgages under specific conditions. If the mortgages are sold, the ABS prototype recalculates possible securitization packages and accompanying risk measures given the number of available mortgages. Afterwards, the ABS package can be sold to interested investors. To illustrate the functional capabilities of the illustrated solution as well as that the above mentioned requirements were satisfied; we tested the prototype using anonymous real-world data. In doing so, we were able to illustrate that it is possible for smaller banks to participate in the international financial market by co-creating securitized ABS bundles.

Through the development of the prototype we present our findings, how smaller banks can benefit from IT services co-creation. To accomplish this goal, Grid computing can provide a viable solution for co-creation of these smaller banks. Controlled from a central IT service provider, applications build upon Grid computing can bundle ABS services jointly together to enable the smaller banks to participate in the ABS market. A single bank in this sector would be too small and has not the appropriate technology in place. However, in order to adopt such a solution, the business value of IT concerning the services sourcing has to be determined. The majority of extant literature on business value of IT has explored IT benefits from the perspective of a single company (or focal firm) [32]. Thereby, the literature focuses solely on increased performances and subsequent cost reductions for the adopting company [26]. Only a few studies considered environmental factors or, more specifically, inter-organizational IT implementations accompanying the business value of IT [26]. The developed prototype follows the call of Kohli and Grover [26, p. 28] who ask for more IT value research in the emerging area of inter-organizational IT value. This can be realized by resources sharing and co-creation. In this paper, we contribute to this area and illustrate how value can be co-created using a securitization-as-a-service solution for smaller banks.

The remainder of this paper is structured as follows: The following section demonstrates the research method with which we developed the prototype together with industry. Next, a literature review on the business value of IT is presented to support our developed prototype in the organizational context. Subsequently, the following section describes the architecture and main characteristics of the implemented prototype for the ABS market. Finally, we conclude with a discussion of the implications for smaller banks and the limitations of this paper.

2 Research Method

DSR provides an appropriate scientific lens for the development of our prototype. The focus of DSR is the creation of IT artifacts and the design of the artificial [36, 38]. Thereby, IT artifacts are either constructs, models, methods, instantiations, or a combination thereof [29]. The prototype for the ABS securities depicts an instantiation.

It encompasses the creation of something new that does not exist so far and that serves human purposes [29]. DSR has its historical origins in the engineering discipline [3, 29] while at the same time being related to many other academic disciplines, e.g., architecture. In both disciplines (i.e. engineering and architecture) as well as in IS, the science of the artificial is of interest to both: science and practice [5]. The theoretical bases for the science of the artificial are so-called kernel theories. Kernel theories result from the evaluation and modification of theories from the natural and social sciences through a creative translation process [21, 31, 38]. The resulting kernel theories form the basis for deriving requirements and the subsequent solution of a real-world problem in DSR [38]. Our prototype builds upon the theoretical basis of the IT business value literature. The developed prototype enables smaller banks to securitize their credits and mortgages on the ABS market. Thereby, the competitiveness of these banks is improved and a common value co-creation is generated. Most DSR researchers would agree that the developed IT artifact by itself may provide a theoretical contribution to the knowledge base provided that key guidelines of DSR are fulfilled and that the given problem is solved [18, 19]. Therefore, a problem-solution and the contribution to the knowledge base is derived either from existing kernel theories and IT artifacts as explained above, or by developing a new IT artifact [23]. In this process, an important initial step is the search for a problem that has practical relevance [21]. In other words, 'a DSR project seeks a solution to a real-world problem of interest to practice' [27, p. 492]. To accomplish this essential goal, it is distinguished between products (IT artifact) and processes (set of activities) in the DSR cycle [29, 38]. The outcome of DSR, i.e., the IT artifact, is always embedded in some place, time, and community and has to be theorized to meet innovative and progressive demands [33].

Concerning the DSR process, it is distinguished between two basic processes, building and evaluating the IT artifact [6, 22, 29]. The first important design process, the building process, is the sequence of activities to produce 'something new', an innovative product. The second important design process, the evaluation process, involves the evaluation of the created IT artifact to provide feedback and relevance to generate new knowledge about the problem at hand. The newly generated insights serve to improve both the quality of the IT artifact and the design process itself [21]. The build and evaluate processes are conducted partly in parallel and involve multiple iterations. Through these multiple iterations, the IT artifact is fully generated to the satisfaction of the researchers and practitioners that later make use of it [31]. Finally, DSR creates a rigorous and meaningful contribution to practice in form of an IT artifact and its evaluation [18]. This paper follows the key guidelines of DSR to develop and evaluate an IT artifact.

3 Business Value of IT Services

Research and theory on business value of IT is a promising meta-theoretical lens through which we regard our DSR approach. Literature on business value of IT is concerned with the return on investment in IT and how IT is improving the production process to an increased economic output [26]. As already mentioned, most research on business value of IT examines on organizational level the company's performance impacts due to IT [e.g., 9, 25, 32]. Most often, IT is regarded either as a variable that is measured through its visible outcomes or as an IT management variable/manifestation that is measured through the manager's decisions [26]. However, the business performance improvements depend on several other factors such as the type of IT, management practices, the organizational structure, or the competitive environment within a company is operating [12, 15]. These endogenous variables do not deal with economic value directly but influence it in the long run [26]. The term "business value of IT" can be used to the company's performance improvement of IT, which includes the productivity enhancement, cost reduction, competitiveness, and other measures of performance [14].

To categorize these different kinds of business value of IT, Melville et al. [32] differentiate IT artifacts and their business value into five categories (see Table 1), which were adapted from Orlikowski and Iancono [33].

Our developed prototype fits to the "tool" category given that the prototype enables the banks to participate in the ABS market and therefore improves the banks competitiveness. Moreover, to analyze the increasing IT value of the company, Melville et al. [32] provided an IT business value model which encompasses the IT's impacts on the business process and the performance. This circumstance is summarized as the "focal firm". However, several other impact factors that are not directly connected to the business value of IT influence the organizational outcome and performance such as the competitive environment [32].

Table 1. IT artifact differentiation used in IT business value research, according to Melville et al. [32]

Category	Description
Tool	IT is a tool intended to generate value, whether productivity enhancement, cost reduction, competitiveness, improved supplier relationships, etc. Specific intention for IT is often unknown. Studies of specific system and implementation contexts enable examination of tool view assumptions.
Proxy	IT is operationalized via proxies such as capital stock denominated in dollars. Wide range of potential proxies exists, but few have been adopted. Adoption of diverse proxies enables triangulation and enhances accumulated knowledge.
Ensemble	Assessment of IT business value generation in rich contexts, often using case or field studies. Organizational structure and co-innovations such as workplace practices may be included as moderators or mediators of value.
Nominal	IT is not conceptualized and appears in name but not in fact. Abstraction enables model precision at the expense of generality.

In summary, the prior literature has explored business value of IT from a single company (focal firm) perspective. Most studies focus on the increased performance or business process outcomes within the company that has invested in IT and do not investigate any spill-over effects on supply chain wide improvements [26]. Only a few studies take also environmental factors and inter-organizational IT implementations into consideration. They discuss outsourcing arrangements [e.g., 16] and how each firm benefits from such a relation but not how small enterprises can cooperatively increase their business value of IT. Thereby, the investment in such a system is a strategic decision since it not only holds the risk of lacking the ability to deal with unforeseen developments occurring outside the companies boundaries but also due to the high investments that have to be made to develop such jointly used systems [7, 28]. Small enterprises differ in terms of their uncertainty regarding IT and competition, limited resources, and their operational focus from larger enterprises [7]. At the same time, they have to be as competitive as larger enterprises in the same area of business which can be achieved by applying an innovative IT strategy [28]. In particular, small enterprises can react faster to changing environments due to flat hierarchies, manageable IT systems and due to their ability to adapt quickly by reorganizing organizational structures [7]. In combination, small enterprises might suffer from limited resources and IT capabilities but can react very flexible to changing environments. It seems feasible to ensure competitiveness by combining their resources and cooperating with other small enterprises while remaining flexible and adaptive at the same time [7]. The developed prototype in this paper meets these criteria allowing small banks to participate in the ABS market while sharing resources and knowledge a single bank alone could not effort or provide.

4 Development of a Prototype for a Cooperative Securitization in ABS Markets

The prototype developed in this paper can be regarded as an innovative solution for the financial services industry to meet the increasing competitive pressure on the financial markets in a collaborative, service-oriented, and cost-efficient way [10, 17, 20, 34].

4.1 Architecture

This section discusses the basic idea of securitization with all involved parties in nine different steps (see Figure 1) [8, 24]. On the one hand, a customer is willing to securitize a credit or mortgage and appears as an originator for this procedure. On the other hand, an investor is willing to invest his or her money in ABS securities. Thereby, the originator pools the demands and sells them to a special purpose vehicle (SPV) which is specially founded for this transaction (step 1). The SPV acts as a broker between these two parties and leads the negotiation. At the same time, an accepted rating agency rates the demands of the originator and attaches the rating to the ABS securities (step 2, 3) which were provided by the SPV (step 4). Hence, the investor receives the conditions under which the securities can be bought and measures the risk of the securities in terms of the rating. If the investor is satisfied, he or she purchases the ABS securities to the presented conditions (step 5). Consequently, the investor pays an emission price to the SPV (step 6). The emission less the originated costs and therewith the emission return is forwarded from the SPV to the originator (step 7). The originator by itself pays an interest and clearance to the SPV for selling the ABS securities (step 8). Finally, the investor receives a coupon and/or clearance from the SPV and receives the ABS securities (step 9).

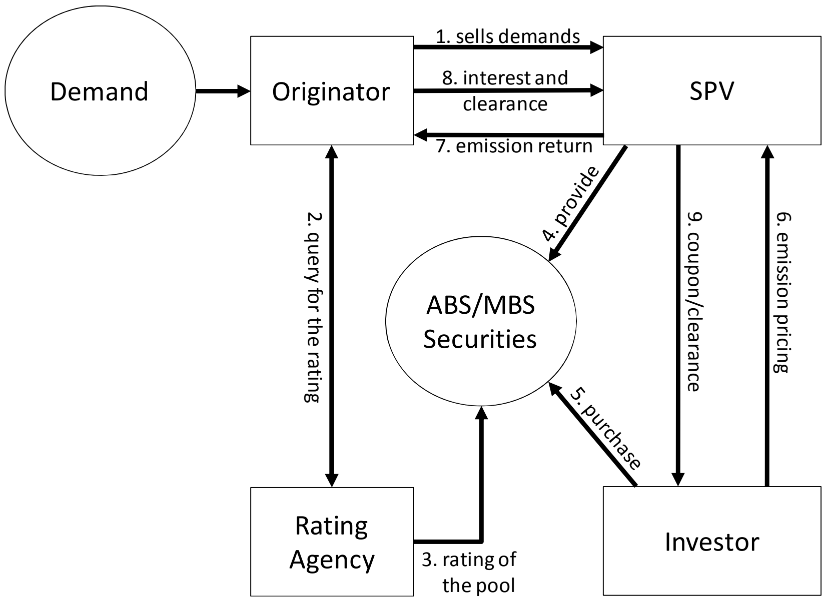


Fig. 1. Overview of the involved parties in an ABS securitization [adapted from 8, 24]

In contrast to the above described securitization, a Grid-based solution provides the opportunity to involve more than one originator in the system and collects as well as measures this information in an effective manner. Given the relatively low mortgage and credit volumes within smaller banks, they are not able to package and sell their securities and thus do not have the opportunity to act on the ABS market. The solution is to pool all mortgages gathered from hundreds of smaller banks, to calculate the attached risk, and finally, to sell the securities to investors [24].

Figure 2 outlines the basic architecture of the developed prototype [adapted from 8, 24]. Basically, the prototype constructs services to select credits or mortgages of the banks by measuring them to the investor's specifications. The credits or mortgages from the small banks are collected from an originating entity (OE). The OE acts as an agent for the participating banks. Within these banks, the managers have now the opportunity to bind their credits or mortgages to different requirements and sell them just under these conditions, whereat these services are combined in a services architecture. However, one of the limitations of the prototype is that in its current form there is no optimization strategy included for the ABS portfolios. By measuring the mentioned information from the banks, an SPV is created especially for this transaction. Another possibility is that the SPV is created by the demand of the investors beforehand. Hereby, the investors have to wait until an OE with appropriate data from the banks is available. However, the SPV together with the OE conclude a credit default swap (CDS) to assure the default risk of the collected credits and mortgages. Moreover, the OE by itself concludes a CDS with every participating bank. Finally, the securities are sold from the SPV to the investors in different tranches (from low to high market risk). The revenues of the sale are not directed to the different smaller banks but rather collected in a collateral pool. The gathering and uniform securitization of the securities do not allow a differentiated allocation of the revenues. The collected revenues are invested in risk neutral securities from the government and allocated to the participating banks. Hence, the banks benefit from the ABS securitization in the following two terms: on the one hand, they have the opportunity to participate in the ABS market which was not possible before and on the other hand, they are able to collect risk neutral securities [8, 24].

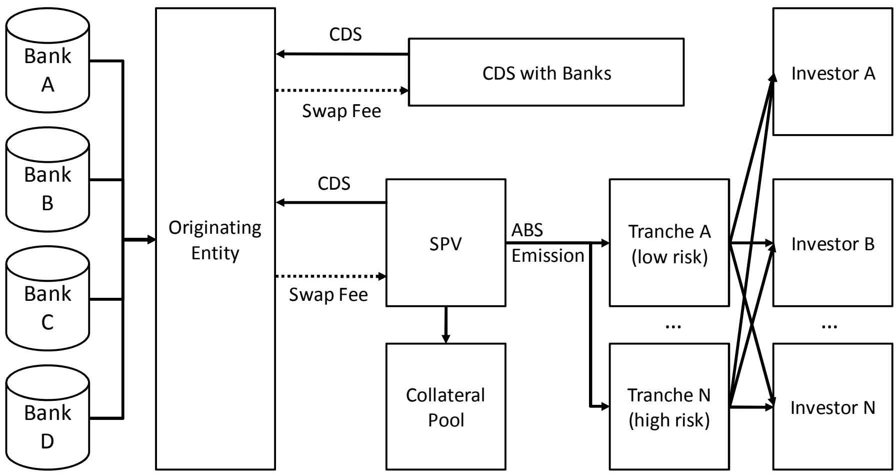


Fig. 2. Overview of the involved parties in an ABS securitization with Grid-technology [adapted from 8, 24]

4.2 Implemented Prototype

The implemented prototype (developed between 2007 and 2009) runs as a web-based Java application and reflects the prior described requirements for securitization on a Grid-engine. The interface is accessible via a secure hash algorithm (SHA) certificate. The prototype itself is hosted in a Grid environment (using Open Source Grid-Middleware Globus Toolkit version 4.x) where it is tested using real-world data. The test data consists of real-world credit and mortgages values from a medium-sized bank and is stored in an Oracle 10g database and accessible via a (4GL) PL/SQL query-language.

The left side of Figure 3 presents a screenshot of the ABS prototype and depicts the different requirements with which the credits or mortgages can be selected from each bank manager, e.g., the duration of the credit or the object type for which the mortgage is inquired. After the selection of the appropriate requirements, the bank receives automatically a menu with credits and mortgages that measure to the prior selected requirements (Figure 3, right side). Thereby, the bank can decide which credits or mortgages should be opened for a securitization by marking the appropriate ones. Thereafter this selection is send to the OE via a service which gathers the information from all participating banks. Hence, the OE either creates a SPV that is

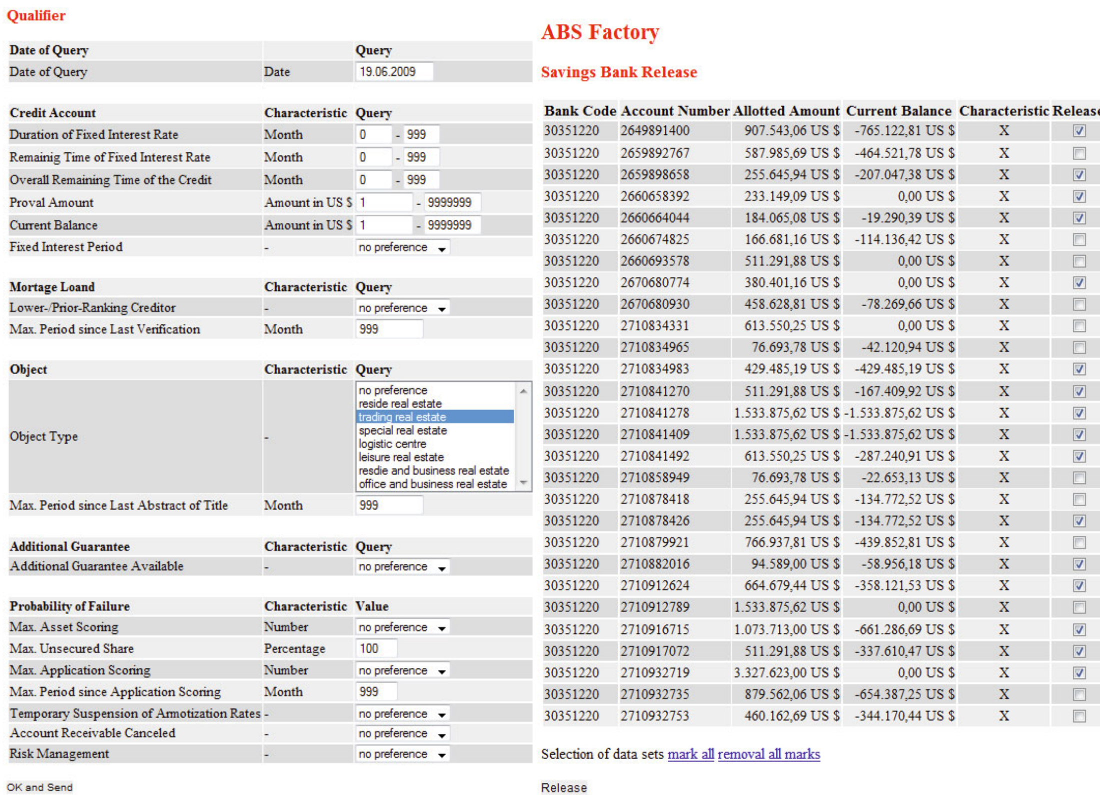
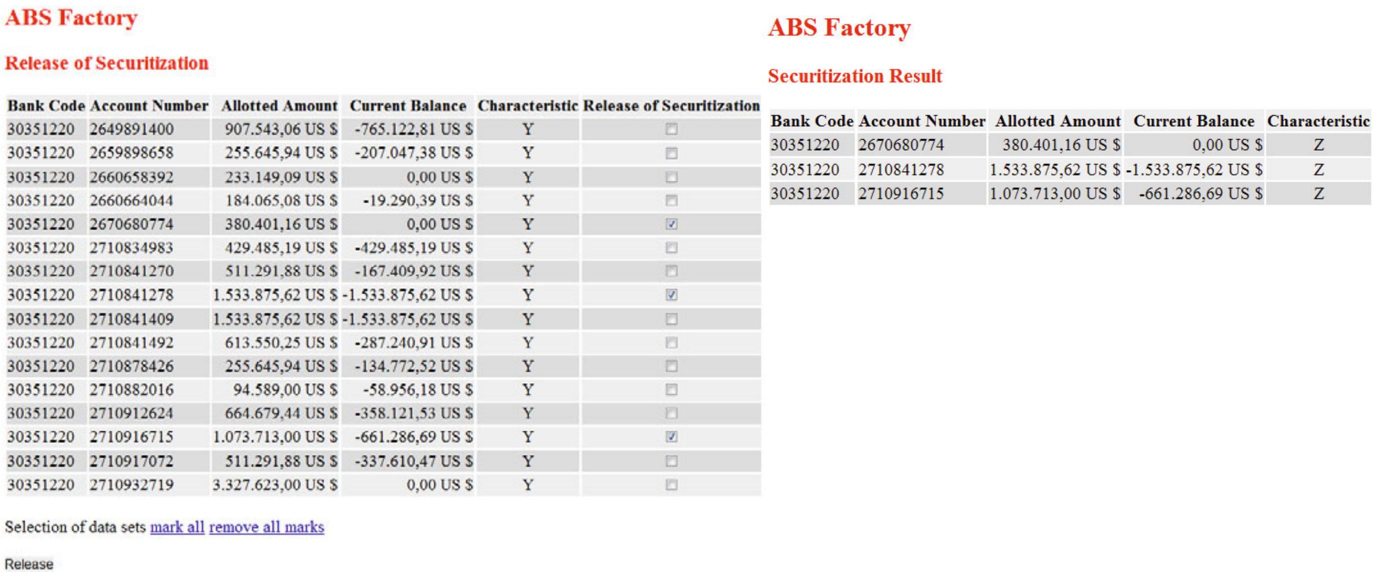


Fig. 3. Screenshot of the ABS prototype. Left: Options of the banks for the securitization of their credits or mortgages. Right: Menu for selecting the credits or mortgages.



ABS Factory

Release of Securitization

Bank Code	Account Number	Allotted Amount	Current Balance	Characteristic	Release of Securitization
30351220	2649891400	907.543,06 US $	-765.122,81 US $	Y	☐
30351220	2659898658	255.645,94 US $	-207.047,38 US $	Y	☐
30351220	2660658392	233.149,09 US $	0,00 US $	Y	☐
30351220	2660664044	184.065,08 US $	-19.290,39 US $	Y	☐
30351220	2670680774	380.401,16 US $	0,00 US $	Y	☑
30351220	2710834983	429.485,19 US $	-429.485,19 US $	Y	☐
30351220	2710841270	511.291,88 US $	-167.409,92 US $	Y	☐
30351220	2710841278	1.533.875,62 US $	-1.533.875,62 US $	Y	☑
30351220	2710841409	1.533.875,62 US $	-1.533.875,62 US $	Y	☐
30351220	2710841492	613.550,25 US $	-287.240,91 US $	Y	☐
30351220	2710878426	255.645,94 US $	-134.772,52 US $	Y	☐
30351220	2710882016	94.589,00 US $	-58.956,18 US $	Y	☐
30351220	2710912624	664.679,44 US $	-358.121,53 US $	Y	☐
30351220	2710916715	1.073.713,00 US $	-661.286,69 US $	Y	☑
30351220	2710917072	511.291,88 US $	-337.610,47 US $	Y	☐
30351220	2710932719	3.327.623,00 US $	0,00 US $	Y	☐

Selection of data sets mark all remove all marks

Release

ABS Factory

Securitization Result

Bank Code	Account Number	Allotted Amount	Current Balance	Characteristic
30351220	2670680774	380.401,16 US $	0,00 US $	Z
30351220	2710841278	1.533.875,62 US $	-1.533.875,62 US $	Z
30351220	2710916715	1.073.713,00 US $	-661.286,69 US $	Z

Fig. 4. Screenshot of the ABS prototype. Left: Menu for securitize the credits or mortgages. Right: Result of the securitization.

especially founded for this transaction or the SPV is created beforehand on the basis of the investors demands. However, the available credits and mortgages from the smaller banks are measured to the investor's demands. The Grid technology mainly supports this process by collecting and measuring the data through the implemented services. A standard server application would not be capable to handle the large amount of data within an acceptable time frame. Thereby, the characteristic X of the right side of Figure 3 represents the current state of the securitization: X means the option to choose which credit or mortgages should be injected into the OE to start the negotiation with the investors; Y means which of the credits or mortgages should be securitized to the negotiated terms; and Z depicts the results of the securitization for the bank.

After matching the data provided by the participating banks with the investor's demands, the prototype sends a selection request back to the small banks with the negotiated requirements of the credits and mortgages (left side of Figure 4).

At this point in time, the SPV acts as a kind of broker and provides the participating banks several options for the securitization of their credits and mortgages. The banks can select which of them to be securitized to the offered conditions. Accordingly, the information of the selected credits or mortgages is gathered again from the OE and directed to the SPV. Finally, the SPV sells them in different risk tranches to the investors. In addition, the small banks receive the result of the securitization of their credits and mortgages (right side of Figure 4).

4.3 Evaluation

The prototype presented in the previous section was evaluated using instruments recommended by DSR [21]. In order to meet changing requirements during the implementation phase of the prototype we adopted the concept of throw away prototypes [31] that helped us to perceive the development process as a creative act of construction and deconstruction rather than designing and delivering an overall satisfying solution in the first attempt. Therefore, we iteratively refined the prototype by permanently adjusting the design and developing process. We used a modular design architecture to support this stepwise refinement of the prototype. 'Experimental' (simulation) and 'testing' (Black Box) evaluating methods were conducted in this creation process [21]. On the one hand, the experimental evaluation included functional tests to discover failures as well as to identify defects and on the other hand, it included the execution of the prototype with real-world data.

Functional tests of the completed prototype were conducted apart from the stepwise evaluation at the beginning of August 2009 which met the expectations of the participating partners [30, 35, 37, 38].

The experimental tests with real-world data were realized by implementing a test program which simulates dozens of smaller banks with different mortgages in stock. The data was provided by a medium-sized bank and depicted their credit and mortgages volume of one year. Table 2 presents the data in detail.

Table 2. Structure of the test data

Type of data	Number of records	Included information
Customer accounts	170,450 records	anonymized account number, anonymized customer number, limit, account balance, period of validity, etc.
Estates	overall more than 670,000 records	period of being customer, type of estate, usage of estate, date of credit application for the estate, local presence of the estate, ranking of the estate, etc.
Credits and mortgages	42,051 records	anonymized account number, amortization rate, interest agreement, interest adaptation, period of validity, amount, etc.
Scoring of the customer	50,000 records	anonymized customer number, date of credit request, score by request, behavioral score of the customer, date of behavioral score, etc.

The simulated banks of the test program injected their assigned mortgages to the prototype with different requirements. The calculations of the prototype were correctly conducted and the securities were sold to the simulated investors. An additional interest was to explore the limitations and performance of the prototype. Unfortunately, the financial crises hindered our industry partners to extend and enlarge the prototype to a real applicable software tool. Therefore, we were not able to test the time delays of the prototype in a productive system. However, through the tests with anonymous real-world data we were able to evaluate the functional readiness of the prototype for the securitization in an ABS market as well as to present the advantages of the participating banks gained through this inter-organizational cooperation.

5 Conclusion

The ABS market is an important source of funding for market participants in the financial services industry. In this market, the smaller banks act in a competitive environment [32] with other larger banks in the financial services industry. Creating an ABS was only possible for these larger banks with a large amount of assets to trade. Smaller banks had not the compute power, amount of mortgages, as well as knowhow to securitize and trade their assets individually. In this paper we presented a prototype that enables smaller banks to create a unique value-creating strategy with respect to their competitors [4, 32] by following a cooperative and inter-organizational strategy [26].

The DSR approach provided us the opportunity to frame the problem beforehand and to develop a potential solution design [23]. To the best of our knowledge, there is no literature on the development and implementation of a prototype that acts as a broker on the ABS market between the involved parties. Moreover, the idea to gather and package the securities of small banks to enable them to participate in this market was not discussed in the previous literature. The majority of extant literature has explored IT benefits from the perspective of a single company (or focal firm) [32]. However, one of the prior articles discussed the differences between an individual and institutional securitization on an ABS market [13]. David [13] found that profits earned from an ABS securitization can be optimized by changing the correlation coefficient between sets of receivables backing different securities for individuals. On the other hand, profits earned from institutions are immune to changes in the correlation and can be controlled only by altering the number of securities created. Other literature concentrates on the role of the ABS market and its impact on the financial crisis [2, 11, 40]. However, none of these articles discuss the problem from a design and developmentalist point of view. In this context we are in line with Holmström et al. [23] who recommended to merge the developmentalist and behavioral parts of the research process together.

IT does not create value for its own sake; there are several factors that are mediating IT and its value creation [26]. For instance, our developed prototype to securitize credits as well as mortgages for ABS markets enables smaller banks to participate in this market but does not provide a business value for its own. Rather the inter-organizational cooperation between these banks to securitize their credits or mortgages provides a kind of co-creation of business value. Hence, the increased business value emerges from the improved competitiveness of the smaller banks by participating in the market [7, 28]. In this context, we followed Kohli and Grovers [26] demand for more research in inter-organizational IT value creation. With the presented solution, the banks can securitize their relatively low credits or mortgages jointly and hand over the risk to the OE and SPV. Thereby, they decrease their bonded capital costs through the securitization with a lower risk by assigning the credits or mortgages to the SPV. In addition, the rating, through an accepted rating agency, of the credits and mortgages become more profitable because of the lower common risk. Hence, the credits and mortgages become more attractive to the investors on the market. This innovative solution improves the competitiveness of the smaller banks through their cooperation [28] and enables them to create an inter-organizational IT value. Otherwise they do not have the level of technical expertise required to develop their own solutions [7].

In summary, we used DSR as an approach to implement a new and innovative IT artifact that presents a solution for an unsolved problem: the missing ABS market-entrance for small banks. The prior discussion outlines the impacts of this solution to the whole market. The small banks are enabled to participate in the market and at the same time improving their competitiveness through this inter-organizational cooperation against larger banks. The outcomes of such an investment are important for a contribution to the IS discipline [1].

To explore the usage, acceptance, and performance of such a system in the daily business, the prototype has to be modified and refined in further research.

References

1. Agarwal, R., Lucas, H.C.: The Information Systems Identity Crisis: Focusing on High-Visibility and High-Impact Research. MIS Quarterly 29, 381–398 (2005)
2. Agarwal, S., Cun, C., DeNardi, M.: Rescuing Asset-Backed Securities Markets. The Federal Reserve Bank of Chicago Essay on Issue, 1–4 (2010)
3. Au, Y.A.: Design Science I: The Role of Design Science in Electronic Commerce Research. Communications of the Association for Information Systems 7, 1–15 (2001)

4. Barney, J.B.: Frim Resources and Sustained Competitive Advantage. Journal of Management 17, 99–120 (1991)
5. Baskerville, R.: What Design Science is Not. European Journal of Information Systems 17, 441–443 (2008)
6. Baskerville, R., Pries-Heje, J., Venable, J.: Soft Design Science Methodology. In: Proceedings of Design Science Research in Information Systems and Technology (DESRIST), Philadelphia, USA (2009)
7. Blili, S., Raymond, L.: Information Technology: Threats and Opportunities for Small and Medium-sized Enterprises. International Journal of Information Management 13, 439–448 (1993)
8. Boehringer, M., Lotz, U., Solbach, C., Wentzler, J.: Conventional versus Synthetic Securitisation – Trends in the German ABS Market. Deloitte & Touche Germany, Duesseldorf, Germany (2001)
9. Brynjolfsson, E., Hitt, L.: Paradox Lost? Firm-Level Evidence on the Returns to Information Systems Spending. Management Science 42, 541–558 (1996)
10. Buhl, H.U., Fridgen, G., Hackenbroch, W.: An Economic Analysis of Service-Oriented Infrastructures for Risk/Return Management. In: Proceedings of the 17th European Conference on Information Systems (ECIS), Verona, Italy (2009)
11. Commission, U.S.S.E.: Dodd-Frank Act Rife with Unintended Consequences. Asset Securitization Report 10, 6–30 (2010)
12. Cooper, B.L., Watson, H.J., Wixom, B.H., Goodhue, D.L.: Data Warehousing Supports Corporate Strategy at First American Corporation. MIS Quarterly 24, 547–567 (2000)
13. David, A.: Controlling Information Premia by Repackaging Asset-backed Securities. The Journal of Risk and Insurance 64, 619–648 (1997)
14. Devaraj, S., Kohli, R.: Performance Impacts of Information Technology: Is Actual Usage the Missing Link. Management Science 49, 273–289 (2003)
15. Dewan, S., Kraemer, K.L.: Information Technology and Productivity: Evidence from Country-Level Data. Management Science 46, 548–562 (2000)
16. Dos Santos, B.L.: Information Technology Investments: Characteristics, Choices, Market Risk and Value. Information Systems Frontiers 5, 289–301 (2003)
17. Foster, I., Kesselman, C., Nick, J.M., Tuecke, S.: Grid Services for Distributed Systems Integration. IEEE Computer 35, 37–46 (2002)
18. Gregor, S.: The Nature of Theory in Information Systems. MIS Quarterly 30, 611–642 (2006)
19. Gregor, S., Jones, D.: The Anatomy of a Design Theory. Journal of Association for Information Systems 8, 312–335 (2007)
20. Leymann, F., Güntzel, K.: The business grid: Providing transactional business processes via grid services. In: Orlowska, M.E., Weerawarana, S., Papazoglou, M.P., Yang, J. (eds.) ICSOC 2003. LNCS, vol. 2910, pp. 256–270. Springer, Heidelberg (2003)
21. Hevner, A., March, S., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28, 75–105 (2004)
22. Hevner, A.R., March, S.T.: The Information Systems Research Cycle. IT Systems Perspective 36, 111–113 (2003)
23. Holmström, J.B., Ketokivi, M., Hameri, A.P.: Bridging Practice and Theory: A Design Science Approach. Decision Science 40, 65–87 (2009)
24. Hull, J.C.: The Credit Crunch of 2007: What Went Wrong? Why? What Lessons Can Be Learned? Journal of Credit Risk 5, 3–18 (2009)

25. Kohli, R., Devaraj, S.: Measuring Information Technology Payoff: A Meta-Analysis of Structural Variables in Firm-Level Empirical Research. Information Systems Research 14, 127–145 (2003)
26. Kohli, R., Grover, V.: Business Value of IT: An Essay on Expanding Research Directions to Keep up with the Times. Journal of the Association for Information Systems 9, 23–39 (2008)
27. Kuechler, B., Vaishnavi, V.: On Theory Development in Design Science Research: Anatomy of a Research Project. European Journal of Information Systems 17, 489–504 (2008)
28. Levy, M., Powell, P.: Information Systems Strategy for Small and Medium Sized Enterprises: An Organisational Perspective. Journal of Strategic Information Systems 9, 63–84 (2000)
29. March, S.T., Smith, G.: Design and Natural Science Research on Information Technology. Decision Support Systems 15, 251–266 (1995)
30. March, S.T., Storey, V.C.: Design Science in the Information Systems Discipline: An Introduction to the Special Issue on Design Science Research. MIS Quarterly 32, 725–730 (2008)
31. Markus, M.L., Majchrzak, A., Gasser, L.: A Design Theory for Systems that Support Emergent Knowledge Processes. MIS Quarterly 26, 179–212 (2002)
32. Melville, N., Kraemer, K., Gurbaxani, V.: Review: Information Technology and Organizational Performance: An Integrative Model for IT Business Value. MIS Quarterly 28, 283–322 (2004)
33. Orlikowski, W.J., Iacono, C.S.: Research Commentary: Desperately Seeking the 'IT' in IT Research – A Call to Theorizing the IT Artifact. Information Systems Research 12, 121–134 (2001)
34. Pollock, J.T., Hodgson, R.: Adaptive Information: Improving Business through Semantic Interoperability, Grid Computing, and Enterprise Integration. John Wiley & Sons, Inc., New Jersey (2004)
35. Pries-Heje, J., Baskerville, R.: The Design Theory Nexus. MIS Quarterly 32, 731–755 (2008)
36. Simon, H.A.: The Science of the Artificial. MIT Press, Cambridge (1969)
37. Srinivasan, A., March, S., Saunders, C.: Information Technology and Organizational Contexts: Orienting our Work along Key Dimensions. In: Proceedings of the International Conference on Information Systems (ICIS), Las Vegas, USA (2005)
38. Walls, J.G., Widmeyer, G.R., El Sawy, O.A.: Building an Information System Design Theory for Vigilant EIS. Information Systems Research 3, 36–59 (1992)
39. Winter, R.: Design Science Research in Europe. European Journal of Information Systems 17, 470–475 (2008)
40. Yeong, C.C., Fitzsimons, A.P., Lange, G.A.: SEC Approves Statement on Global Accounting Standards and Proposes Rules for Asset-Backed Securities. Bank Accounting & Finance 23, 37–42 (2010)

Service Extraction from Operator Procedures in Process Industries

Jingwen He[1], Sandeep Purao[1], Jon Becker[1], and David Strobhar[2]

[1] College of Information Sciences and Technology, The Pennsylvania State University, 16802 PA, United States
{jxh497,spurao,jdb373}@ist.psu.edu
[2] Beville Engineering, Inc. 45459 OH, United States
dstrobhar@beville.com

Abstract. Procedures are a common knowledge form in process industries such as refineries. A typical refinery captures hundreds of procedures documenting actions that operators must follow. Maintaining the action-knowledge contained in these procedures is important because it represents a key organizational asset that can be leveraged to minimize the threat of accidents. We develop an approach that extracts services from these operator procedures. The paper describes the heuristics underlying this approach, illustrates its application, and discusses implications.

Keywords: Service Extraction, Knowledge Modules, Knowledge Representation, Heuristics.

1 Introduction

Procedures are pervasive across process industries [1-4]. They cover activities varying from the routine to the rare, with time spans ranging from minutes to days. A petrochemical plant, for example, can contain hundreds of documented procedures. These procedures present two significant problems. First, in the absence of industry standardization, they represent large chunks of unstructured knowledge that is rife with unnecessary duplication. Second, they contain instructions that assume uniform levels of expertise in the audience. The first problem requires a solution to actively extract and manage knowledge contained in these procedures [5]. The second problem points to an assumption that is increasingly untenable because of a looming wave of retirement among operators in the petrochemical industry [6]. This research is motivated by the above problems, using the oil refining domain, treating as a domain representative of many process industries.

We adopt the position suggested by Bera and Wand [7] that the instructions contained in operator procedures represent action knowledge. Extending this position, we re-conceptualize each procedure as a set of services, consisting of a well-defined set of instructions. Extracting these services to manage the action knowledge requires a concerted effort. In this paper, we propose such an approach. It consists of heuristics to extract these services. The anticipated outcome of our approach is a set of services, each containing an action knowledge module that may be independently maintained, manipulated, and tailored for different audiences.

H. Jain, A.P. Sinha, and P. Vitharana (Eds.): DESRIST 2011, LNCS 6629, pp. 335–349, 2011.

The approach we develop consists of heuristics that leverage structural properties of the procedures, coupled with a lightweight ontology, i.e., domain expertise contributed by subject-matter experts from the petrochemical industry. The paper reviews prior work in section 2. In section 3, we describe the approach and develop the heuristics. Section 4 illustrates application of the heuristics to the procedures. We conclude in section 5 with a discussion of implications, limitations and future work.

2 Background and Review of Prior Work

Reports indicate that major industrial accidents (such as those in petrochemical plants and refineries) are less likely today than they were three decades ago [8]. Significant strides have been made in organizational practices, training routines and installing a safety culture in refineries [9-11]. These, in turn, have resulted in increased levels of expertise among operators and have contributed to a decrease in accident rates. Accident tracking data and preliminary analyses, however, continues to implicate causes such as lack of (availability and use of) effective procedures, and lack of (or inadequate) training [12]. The scale of the problem is critical for the petrochemical industry. Accidents, if they happen, can affect not just property but also cause loss of lives; and can damage not just the industrial assets but also communities surrounding the refinery. There are about 140 refineries in the U.S., each with about 250 operators[1]. During their tenure at the refinery, the operators acquire and cultivate expertise that becomes hard to acquire for newcomers [13]. The looming wave of retirement is expected to reduce the ranks of expert operators in the petrochemical refineries by as much as 20% in the next several years [13]. This research represents a potential response to the train-wreck scenario anticipated by these trends.

2.1 An Action View of Knowledge and Procedures

In anticipation of the problems identified above (and as the industry has learned more about the work that operators do), the petrochemical industry has taken steps to codify a number of procedures. These procedures are used in most, if not all, refineries to provide oversight and control of operator performance [14]. Although their format can vary, they fundamentally contain sets of instructions that operators must follow [12], sometimes with additional information such as underlying rationale. These sets of instructions represent knowledge for action that operators use. This ‘action’ view of knowledge has roots in prior work in artificial intelligence [15, 16]. It suggests an emphasis that is different from the truth-value of knowledge and provides an important precursor to our work [17].

Our conceptualization closely follows the ‘action’ view of knowledge suggested by Bera and Wand [7]. They build on the precursors mentioned above to argue that it is effective performance, not justified true belief, that defines the basis for knowledge [7]. Blosch [17] suggests a complementary view, emphasizing the role of ‘action’ knowledge in ensuring the successful accomplishment of practice. This ‘action’ knowledge may be expressed as instructions, steps or manuals, that is, as explicit

[1] Number and Capacity of Petroleum Refineries:
http://www.eia.doe.gov/dnav/pet/pet_pnp_cap1_a_(na)_8OO_Count_a.htm

knowledge [18]. Our conceptualization of operator procedures as 'action' knowledge, thus, represents an application and extension of the ideas suggested by Newell and Simon[16], Nonaka[18], and Bera and Wand[7]. The extension we suggest involves re-conceptualizing a procedure as a set of services, which are larger granularity than individual instructions. We elaborate on this next.

2.2 Service Extraction via Procedure Modularization

Procedures implement the action view of knowledge outlined above. A typical refinery includes hundreds of such procedures. Each codifies a complex set of instructions to achieve a larger goal such as start-up or shut down (e.g. a procedure to perform a controlled shutdown for the Hydro-cracking unit). Several factors suggest that decomposing each procedure into services – service extraction [19] – is likely to be appropriate for these long and complex procedures. First, different operators (such as field or console operators) may be responsible for carrying out different subsets of instructions. Second, different parts of the procedures may be carried out at different times. Third, some instructions may require operating in different locations or on different equipment within a refinery. Finally, there may be sub-sets of instructions that are common across multiple of procedures.

The term 'service extraction' was originally coined in the context of legacy systems to describe the identification of smaller-grain, self-contained units from larger, monolithic systems [20, 21]. An extracted service describes a set of operations that may be independently deployed, invoked and monitored [22]. In the context of operator procedures, we define service extraction as the extraction of self-contained components of action knowledge that may be communicated and used by operators.

It is, however, difficult to disentangle the pieces of action-knowledge that are embedded in the operator procedures because the instructions are in natural language and may require significant domain knowledge. To achieve this goal, we plan to draw on a number of fundamental attributes of instructions such as actor switches, temporal interruptions, and location movements. Each presents an opportunity to define boundaries around a subset of instructions. However, it also presents a challenge because of the scale of the problem: hundreds of long procedures in each refinery with ongoing updates to each. In the next section, we develop an approach for service extraction that is amenable to automation.

3 Service Extraction from Operator Procedures

The approach we propose for service extraction from operator procedures is driven by the considerations outlined in the review of prior work. First, we acknowledge that operator procedures contain action-knowledge [7] written in natural language that must be parsed before identifying services. Second, we acknowledge the importance of domain knowledge in this parsing task. Third, overlaps across procedures suggest that these commonalities can be explored to identify common services across procedures. Fourth, procedures such as different operators, locations and time-spans provide possible leads for identifying boundaries around services. Because each of these strategies is unlikely to produce an error-free set of outcomes, we argue for a heuristic [23], instead of algorithmic, approach for service extraction from operator procedures. We define a service as a self-contained subset of instructions, extracted from one or more procedures [20, 21], that is a directly addressable and editable chunk of action knowledge.

The proposed service extraction approach consists of three phases. The first identifies the action knowledge component in the procedures by removing the pre-amble. The second parses each instruction in the action knowledge component to identify important elements. The final phase identifies appropriate subsets of instructions, and extracts these as services. Figure 1 outlines the approach.

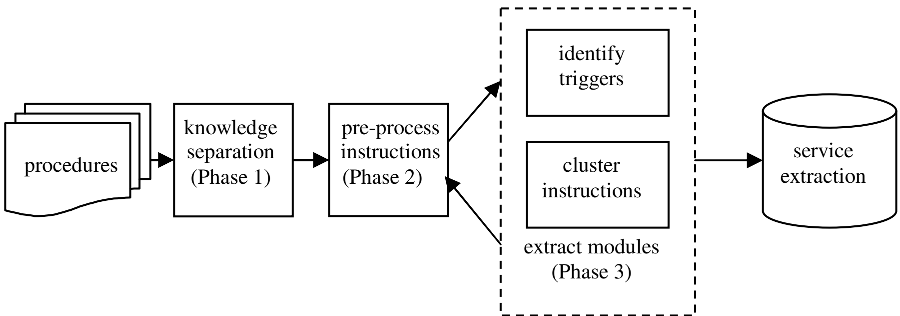


Fig. 1. A Heuristic Approach to Service Extraction from Operator Procedures

We conceptualize each procedure as a set of statements: some containing instructions, i.e. action knowledge, others containing information such pre-amble. The heuristics analyze the instructions by leveraging their structural, syntactic and semantic attributes [24]. The heuristics first use part-of-speech tagging [25] to identify elements in each instruction such as actions, actors and objects. These are then subjected to heuristics to identify and extract services. Table 1 outlines the notations.

Table 1. Notations

Set Variables	
S_n:	the n-th line of procedure statement
K_i:	the i-th keyword of statement S_n, $k_i \in S_n$
v_i:	the i-th verb in list V
n_i:	the i-th noun in list N
Lightweight ontology of domain concepts	
TitleIndicator:	list of labels that are indications of title
V:	list of verb or derived phrases denoting predicates
N:	list of noun or derived phrases and abbreviations denoting refinery processes
Conj:	list of noun or derived phrases denoting conjunctions
EndingList:	list of routine actions to terminate procedures
WaitingList:	list which contains any phrase indicating break in time
LocationList():	lists of units grouped based on their physical distances

Table 1. (*continued*)

OutsideActor	list of noun or derived phrases denoting console operators
InsideActor	list of noun or derived phrases denoting field operators
Indicator variables	
S_n.title:	flag to record whether S_n is a part of title
S_n.meta-info:	flag to record whether S_n is a statement that contains meta-information
S_n.maintext:	flag to record whether S_n is a statement contained in the main body of the procedure
S_n.predicate:	the predicate of S_n which denotes action
S_n.actor:	the subject of S_n which denotes the actor
S_n.object:	the object of S_n which denotes the target plant where action is applied to
S_n.condition:	the conjunction of S_n which denotes the beginning of condition
S_n.StartTrigger:	flag to record whether S_n is the start trigger of the procedure
S_n.EndTrigger:	flag to record whether S_n is the end trigger of the procedure
S_n.BreakByActor:	flag to record whether there is a procedure break because of actor change at S_{n+1}
S_n.BreakByTiming:	flag to record whether there is a procedure break because of waiting condition S_n
S_n.BreakByLocation:	flag to record whether there is a procedure break because of location change of S_n.object
S_n.BreakByCoocurrence:	flag to record whether there is a procedure break because S_n and S_{n+1} seldom co-occurr
Thresholds	
Relation[$(v_i, n_i) \times (v_j, n_j)$] :	occurrence of two actions (v_i, n_i) and (v_j, n_j) in an order
Threshold:	to indicate whether the two instructions should be separated
Functions	
MetaInfoLocate():	a function to identify location of meta information, given the location of title
MainTextLocate():	a function to identify location of main body, given the location of title
PhraseAfterConjunction():	a function which return the whole phrase after the input conjunction
POST():	a function which calls part-of-speech-tagging process

3.1 Heuristics for Phases 1 and 2: Separating Content and Identifying Elements

The first phase is supported by three heuristics that parse each procedure to separate the action-knowledge content. The first heuristic scans the procedure to locate its title. A taxonomy of equipments and units in the refinery aids this heuristic. Based on the position of the title, the meta-information and main body of the procedure (containing action knowledge) is located relative to the position of title by heuristics 2 and 3. Table 2 summarizes these heuristics.

Phase 2 pre-processes the main body of the procedure. This part of the procedure is conceptualized as a set of instructions, i.e., action knowledge. The pre-processing is accomplished by extending part-of-speech-tagging (POST) [25] with terms from the

Table 2. Heuristics to Separate Action Knowledge Content in Operator Procedures

Heuristic	Assumptions and Formalization
Heuristic 1 – Title Identification	Assumption: The Title contains a Term that describes the process affected. $\exists\, k_i \in S_n \mid k_i \in$ TitleIndicator $\Rightarrow S_n$.title = 1
Heuristic 2 – Meta-Info Identification	Assumption: Meta-information is relative to the position of the Title (see Heuristic 1). $\forall S_n \mid (S_i$.title = 1) $\wedge$ ($S_n \in$ MetaInfoLocate(S_i)$\Rightarrow S_n$.meta-info = 1
Heuristic 3 – Body of Procedure Identification	Assumption: The location of the main body of the procedure is relative to the position of the Title (see Heuristic 1). $\forall S_n \mid (S_i$.title = 1) $\wedge$ ($S_n \in$ MainTextLocate(S_i) $\Rightarrow S_n$.maintext = 1

lightweight ontology that reflect domain knowledge along with a dictionary that differentiates action, such as predicates, subjects, objections, and conjunctions [26]. The heuristics in this phase allow re-structuring each instruction into action descriptions made up of a responsible actor, the subject of action and a verb that indicates the action. Table 3 summarizes the four heuristics that support this phase.

Table 3. Heuristics for Pre-processing of each Instruction in the Procedures

Heuristic	Assumptions and Formalization
Heuristic 4 – Predicate Tagging	Assumption 1: Each instruction contains at most one predicate. Assumption 2: The predicate denotes the action of this instruction. ($\forall k_i \in S_n \mid$ POST(k_i) $\in$ V) $\Rightarrow S_n$.predicate = k_i
Heuristic 5 – Action Object Tagging	Assumption 1: The object of instruction is identified from a taxonomy. Assumption 2: Objects appear after verbs. ($\forall k_i \in S_n \mid$ POST(k_a) $\in$ V $\wedge$ POST(k_i) $\in$ N $\wedge$ i > a) $\Rightarrow S_n$.object = k_i
Heuristic 6 – Actor Tagging	Assumption 1: There is at most one actor for each instruction. Assumption 2: Two types of operators: inside (console) and outside (field). ($\forall k_i \in S_n \mid$ POST(k_a) $\in$ V $\wedge$ POST(k_i) $\in$ ActorOfInsider) $\Rightarrow S_n$.actor = insider ($\forall k_i \in S_n \mid$ POST(k_a) $\in$ V $\wedge$ POST(k_i) $\in$ ActorOfOutsider) $\Rightarrow S_n$.actor = outsider
Heuristic 7 – Condition Tagging	Assumption: Multiple instructions are connected via conjunctions. ($\forall k_i \in S_n \mid$ POST(k_i) $\in$ Conj) $\Rightarrow S_n$.condition = PhraseAfterConjunction(k_i)

3.2 Heuristics for Phase 3: Identifying and Extracting Modules as Services

The third phase is supported by six heuristics that use the pre-processed instructions to identify and extract modules as services. They are explained next with the help of short examples. The first two heuristics identify start and end triggers.

Heuristic 8. Start Trigger Identification. This heuristic identifies the trigger that initiates the sequence of instructions specified in the Procedure. It is often specified as a pre-condition, i.e., without an explicit action. It describes either a value (range) for a parameter of interest or elapsed time (range) that, once breached, requires that the operator to initiate the Procedure. A lightweight ontology that describes the key parameters is, therefore, essential for recognizing the start trigger. For example, a start trigger may be a description of failure such as “fuel gas failure”, “loss of instrument air”, or a condition such as “when the instrument air failure falls to 16.5 psig.” It is important to identify the Start Trigger because it provides an explicit condition that may invoke multiple possible services that initiate different procedures.

Heuristic 9. End Trigger Identification. This heuristic identifies the trigger that specifies the wrap-up activities such as reporting results or checking parameters. They are often common across multiple procedures and can, therefore, appear as services that are shared across procedures. A lightweight ontology of terms, in this case, one of the actions that indicate that the wrap-up actions have begun, is essential to identify the end trigger. Phrases such as “notify process manager,” “notify duty man,” and “review unsatisfactory readings with head operator” are examples of commonly occurring review and reporting actions at the end of procedures. Identification of the end trigger and consequently, the wrap-up service, is important because it allows modularizing a set of instructions that can be reused across multiple procedures.

Four other heuristics cluster procedure instructions to identify and extract services. The inputs to these heuristics are the pre-processed instructions that remain after the modules identified by the start and end triggers are removed.

Heuristic 10. Time-based Clustering. This heuristic clusters instructions within a procedure that occur without a break in action. These are instructions that are not interrupted by conditions such as waiting for an external action or till a certain parameter value is reached. These instructions may be performed by one or more operators, in sequence or in parallel without an apparent pause. The basic idea underlying this use of activity timing comes from studies of event-condition-action models [27]. An interruption in the procedure serves as the boundary around the service defined by this cluster of instructions. Here as well, a lightweight ontology of terms is essential. We use indicators of interruptions obtained from the domain experts for this purpose. For example, indicators such as “wait” and “till” provide clues about the boundary around an instruction cluster. Such time-based clustering is important because it provides a first clue to decomposing the procedures into modules that can lead to service extraction.

Heuristic 11. Location-based Clustering. This heuristic clusters the instructions within a procedure based on the physical proximity of their target objects. For example, a procedure may contain contiguous instructions but one may require action such as “turning a valve” in one location, followed by “checking a meter” in another location. If these two locations are far, then it is an indication that these two instructions, in spite of their apparent sequence without pause within the procedure, belong to different modules. Operationalizing this heuristic requires non-trivial information about the geo-locations of different target objects. With the help of such geo-locations, it is, however, straightforward to identify boundaries around instruction sets that allow such separation among instruction clusters. Such location-based clustering is important because it provides additional clues, beyond those provided by time-base clustering, for identification and extraction of services.

Heuristic 12. Actor-based Clustering. The Actor responsible for each instruction provides another clue to instruction clustering. This is true because many procedures are complex and require coordination across multiple operators. For example, a procedure to shutdown a unit may require field operators to carry out some instructions (such as "closing a valve"), and the console operators to follow others (such as "monitoring a parameter"). Implementing this heuristic requires examining the Subject for each instruction in the procedure. A change in the subject suggests the boundary around a cluster of instructions. Such actor-based instruction clustering is important because it can produce services that have clear assignment of responsibility.

Heuristic 13. Co-occurrence-based Clustering. Unlike the previous three heuristics that leverage a lightweight ontology such as the location information, time indication, and actor types, this heuristic relies on a statistical pattern. It tries to identify co-occurring pairs of instructions in multiple procedures. The relative frequency of co-occurrence is used by the heuristic to identify and extract this cluster of instructions as a service. A key decision for this heuristic is the threshold frequency, which we allow to be user-determined. The significance of this heuristic is straightforward. It allows extraction of repeating clusters of instructions into modules that can be extracted as services. Table 4 summarizes the heuristics.

Table 4. Heuristics for Identifying and Extracting Services from Procedures

Heuristic	Assumptions and Formalization
Heuristic 8 – Start Trigger Identification	Assumption: The start trigger is the first instruction with no explicit action. $\exists S_n \mid S_{n-1}.\text{maintext} != 1 \wedge S_n.\text{maintext} = 1 \wedge S_n.\text{predicate} = \text{null} \Rightarrow S_n.\text{StartTrigger} = 1$
Heuristic 9 – End Trigger Identification	Assumption 1: Main body text of procedure ends with instructions. Assumption 2: The end trigger is the last action. Assumption 3: The end trigger is one of special actions in EndingList. $\exists S_n \mid S_{n+1}.\text{predicate} = \text{null} \wedge S_n.\text{predicate} \in \text{EndingList} \Rightarrow S_n.\text{EndTrigger} = 1$
Heuristic 10 – Time-based Clustering	Assumption: A break in activity e.g. checking/waiting suggests a boundary. $\forall S_n \mid S_n.\text{condition} \in \text{WaitingList} \Rightarrow S_n.\text{BreakByTiming} = 1$
Heuristic 11 – Location-based Clustering	Assumption: Actions on units in proximity belong to the same module. $\forall S_n \mid S_n.\text{object} \in \text{LocationList(j)} \wedge S_{n+1}.\text{object} \mathrel{!\in} \text{LocationList(j)} \Rightarrow S_n.\text{BreakByLocation} = 1$
Heuristic 12 – Actor-based Clustering	Assumption: A switch inside to outside or vice versa suggest a break. $\forall S_n \mid S_n.\text{actor} != S_{n+1}.\text{actor} \Rightarrow S_n.\text{BreakByActor} = 1$
Heuristic 13 – Co-occurrence-based Clustering	Assumption: High co-occurrence of two actions suggests membership in a module. $(\forall v_i, v_j \in V, \forall n_i, n_j \in N \mid S_{n-1}.\text{predicate} = v_i \wedge S_{n-1}.\text{object} = n_i \wedge S_n.\text{predicate} = v_j \wedge S_n.\text{object} = n_j) \Rightarrow \text{Relation}[(v_i, n_i) \times (v_j, n_j)] ++$ $\forall S_n \mid \text{Relation}[(S_n.\text{predicate}, S_n.\text{object}) \times (S_{n+1}.\text{predicate}, S_{n+1}.\text{object})] \leq \text{threshold} \Rightarrow S_n.\text{BreakByCoocurrence} = 1$

The heuristics identify instruction sets as modules and extract these as a first approximation of services. The next section illustrates their use with real-world procedures obtained from multiple refineries.

4 Application and Evaluation

The heuristic approach we have outlined is difficult to evaluate because of a number of concerns. First, the heuristics must be tailored to differences in procedure formats from different organizations because the petrochemical industry does not follow a standard format for documenting procedures. Second, the application of heuristics is data-driven, i.e., the applicability of heuristics to procedures from different petrochemical companies is likely to be different. Third, statistical pattern mining across multiple procedures can be difficult to demonstrate because of the availability of a small number of procedures.

The application and evaluation we report is, therefore, formative, instead of summative evaluation [28]. The approach we follow for evaluation resembles a case-based method that focuses on illustration [29, 30] following a descriptive approach [30]. The example procedure(s) we use for evaluation describes a set of instructions for failure recovery at the Hydrocracker unit at a refinery (anonymized). The procedure is named DHT Shutdown Procedure – Fuel Gas Failure. It contains 33 statements. The description and illustrations show outputs obtained by applying the heuristics along with the rationale. Before proceeding to the illustration, we describe the lightweight ontology along with examples. It consists of taxonomies of terms, including actions, actors, equipments and locations obtained from subject-matter experts. Table 5 shows the categories and examples in each.

Table 5. Lightweight Ontology and Example Elements

Category	Examples
Phrases indicating Title	standing instruction, DHT shutdown procedure, col 13 …
Verbs indicating Actions	close, shut down, open, block off, start, steam …
Nouns indicating Objects	feed pump, damper, accumulator, Htr 30, CCU …
Conjunctions	when, till, unless …
Procedure End Actions	notify process manager, notify duty man, fill in report …
Wait conditions	oil from Htr 29 reaches 450^{o}F, reactor pressure at 200 psig …
Labels for Inside Actor	insider, I, console, console operator
Labels for Outside Actor	outsider, O, field, field operator
Geo-coded Locations	Feed heater and reactor: Htr 29, heater 30, feed pump, feed line

The heuristics in Phase 1 separate the procedure into three parts: meta-information, title, and the main body of the procedure (see Table 6). The first heuristic uses (a) a key phrase indicating “title”, and (b) a label indicating equipment. Together, they allow marking of a part of the procedure as the title. Based on the location of the title (in this procedure, at Statements 5 and 6), the location of meta-information and the main body of the procedure (containing action knowledge) are determined by Heuristic 2 and 3 based on the position of the text relative to the title. The main body of the procedure contains 27 statements, which are then subjected to heuristics from the later phases.

Table 6. An illustration showing application of Phase 1 heuristics

Statement	Body of Procedure	Output of Phase 1
S_1	Reference No.: DDHE – 9	S_1*.meta-info = 1*
S_2	Page: 1 – 1	S_2*.meta-info = 1*
S_3	Date: MM/DD/YY	S_3*.meta-info = 1*
S_4	By: xxxxx	S_4*.meta-info = 1*
S_5	STANDING INSTRUCTION NO. DDHE – 9	S_5*.title = 1*
S_6	DHT SHUTDOWN PROCEDURE – FUEL GAS FAILURE	S_6*.title = 1*
S_7	Fuel gas failure	S_7*.maintext = 1*
S_8	Close main block on fuel and pilot gas lines	S_8*.maintext = 1*
….	….	….

In Phase 2, instructions in the body of the procedure, that is instructions 7 through 33, are parsed using part-of-speech-tagging [25]. The heuristics identify and tag the following elements in each instruction: predicate, subject, object, and condition. The process is augmented with the lightweight ontology described above. Table 7 illustrates the outcomes for some of the instructions.

Table 7. An illustration showing application of Phase 2 heuristics

	Body of Procedure	S_i. subject	S_i. predicate	S_i.object	S_i. condition
S_7	Fuel gas failure	*field*	*null*	*null*	
S_8	Close main block on fuel and pilot gas lines	*field*	*close*	*fuel and pilot gas lines*	
S_9	stream to Htr 29 & Htr 30 fireboxes	*field*	*steam*	*Htr 29 & Htr 30*	
S_{10}	Open dampers	*field*	*open*	*dampers*	
S_{11}	Divert stripper bottoms back to feed	*field*	*divert*	*stripper bottoms*	
S_{12}	Start electric pump	*field*	*start*	*electric pump*	
S_{13}	Circulate stripper bottoms	*field*	*circulate*	*stripper bottoms*	
S_{14}	Shut down power recovery turbine	*field*	*shut down*	*power recovery turbine*	
S_{15}	Shut down feed pump	*field*	*shut down*	*feed pump*	
S_{16}	If feeding USC, notify the CCU	*field*	*notify*	*CCU*	*feeding USC*
S_{17}	Cut out all USC	*field*	*cut out*	*USC*	
S_{18}	Close annin valve in feed line	*field*	*close*	*feed line*	
S_{19}	Shut down field feed pump	*field*	*shut down*	*field feed pump*	
S_{20}	Block off platformer	*field*	*block off*	*platformer*	

Table 7. (*continued*)

S_{21}	unload compressor make valves	*field*	*unload*	*compressor make valves*	
S_{22}	Notify the SMR that H2 is not longer needed	*field*	*notify*	*SMR*	
S_{23}	Shut down condensate injection	*field*	*shut down*	*condensate injection*	
S_{24}	Shut down sour water pump	*field*	*shut down*	*sour water pump*	
S_{25}	Close in warm and cold flash accumulators with normal levels	*field*	*close*	*warm and cold flash accumulato rs*	
S_{26}	Close in lean and fat DEA circulation	*field*	*close*	*lean and fat DEA circulation*	
S_{27}	Shut down Htr30 circulation	*field*	*shut down*	*Htr30*	
S_{28}	Pump stripper bottoms back to feed	*field*	*pump*	*stripper bottoms*	
S_{29}	As time permits close individual fuel	*field*	*close*	*individual fuel*	*time permits*
S_{30}	close pilot burners	*field*	*close*	*pilot burners*	
S_{31}	Shut down feed and product inhibitor injection	*field*	*shut down*	*feed and product inhibitor injection*	
S_{32}	Shut down air fans	*field*	*shut down*	*air funs*	
S_{33}	If unable to restart unit, notify Process Manager	*field*	*notify*	*Process Manager*	*unable to restart unit*

The heuristics in Phase 3 use the structured instructions (see Table 7) produced by the last phase. Heuristic 8 is triggered by the Null predicate in instruction 7 (first line in Table 7). It sets this statement as a start module. Heuristic 9 is triggered by the predicate "notify" in instruction 33 (last line in Figure 3). Heuristic 10 is not triggered because none of the predicates indicating a time interruption are found in the instructions. Heuristic 11 is triggered based on information about location of units. The instructions show that the unit changes twice during the procedure. Each change triggers Heuristic 11 to set a procedure break. In the example procedure, the target object of instructions 8 to 11 appear in the same location (obtained from the lightweight ontology). The target object of instruction 12 is not co-located with the target object for instructions 8 to 11. This location change causes Heuristic 11 to set a break between statements 11 and 12. Heuristic 12 is not triggered either because all instructions are carried out by the same operator role, the outside, field operator. Heuristic 13 relies on mining co-occurrence of patterns across multiple procedures. The example procedure shown cannot illustrate this heuristic. After applying the heuristics, the procedure is decomposed into multiple modules, each defined as a service. The services extracted are shown in Table 8 below.

Table 8. Services extracted from the Procedure

Cluster	Contents	Service Description
S_7	Instruction: Fuel gas failure	*Start Trigger Service*
S_8 to S_{19}	Starting from "Close main block on fuel and pilot gas lines" to "Shut down field feed pump"	*Service 1: 12 instructions* *Primary Operator – Field Operator* *Location – Feed Heater and Reactor*
S_{20} to S_{26}	Starting from "Block off platformer" to "Close in lean and fat DEA circulation"	*Service 2: 7 instructions* *Primary Operator – Field Operator* *Location – Recycle Compressor*
S_{27} to S_{32}	Starting from "Shut down Htr30 circulation" to "Shut down air fans"	*Service 3: 6 instructions* *Primary Operator – Field Operator* *Location – Feed Heater and Reactor*
S_{33}	Instruction: If unable to restart unit, notify Process Manager	*End Trigger Service*

The illustration provides prima facie validation of the approach by showing the ability to parse the 33 instructions in the procedure to extract five services (See Table 8).

5 Conclusions and Future Work

This research has proposed a heuristic approach to extract services that represent action knowledge modules from procedures in a petrochemical refinery. The approach is inspired by work related to an action view of knowledge [7, 15, 16]. The service extraction approach identifies and extracts services from operator procedures. In doing so, we have extended the service extraction perspective [20, 21] in a novel manner – applying it to operator procedures in a process industry instead of its traditional application to extraction of services from legacy systems.

We claim several potential benefits that follow from our approach. First, the approach exposes fundamental properties of each instruction such as timing, location, and subject (actor) in a manner that make each instruction amenable to manipulation. Second, it is likely that the services extracted with our approach are easier to manage and reuse across multiple procedures. Third, the availability of extracted services provides more freedom for tailoring the presentation to operators who may exhibit different levels of expertise. Fourth, the services extracted can be used to drive training efforts for novices as well as experts who may make delegation decisions.

We acknowledge that there are some limitations to our approach. First, the emphasis on action knowledge structured as predicates, subjects, objects, and conditions may lead operators to ignore descriptive or peripheral knowledge that may be implicitly embedded in instructions. Second, services may emphasize the "how" instead of "why" slowing the development of experience-based expertise. Although we indicate this as a potential threat, it is possible to supplement the services with a conceptual map of the refinery and the rationale that enhances the operators' understanding. Third, our approach requires some extra efforts to create a domain-specific lightweight ontology from subject-matter experts. The effectiveness of services is dependent on the quality and veracity of information in this lightweight ontology. Fourth, it is essential to understand if there is any knowledge that captures dependencies across services which is lost via our approach. We plan to investigate this concern as we continue our research. Finally, the service extraction approach is based on existing procedures. It does not immediately lead to the identification of new procedures. Here as well, we will explore the possibility of combining services to identify new procedures as the research progresses.

The work we have described for service extraction relies on heuristics. We have argued that this is an appropriate approach because of the nature of knowledge in the procedures. Nevertheless, we acknowledge that there are disadvantages to using a heuristic approach. These include the following. First, the heuristics require domain expertise that may be difficult to obtain. Second, it is difficult to claim comprehensiveness. There may be additional heuristics that we have not discovered yet. Third, the sequence in which the heuristics are applied may influence the outcomes. In particular, it is necessary to account for potential interactions among the heuristics themselves. These remain on our future research agenda.

In spite of the limitations acknowledged above, the approach has considerable potential for advancing the state of the practice. Possible applications of such an approach can be found in other industries with large amount of procedures, instructions, and working guidelines. Examples include monitoring nuclear plants [31] and healthcare procedures [7]. Although prior work has suggested possibilities for automating procedures in process industries, the human-in-the-loop phenomenon is increasingly recognized as critical [32, 33]. Improving this aspect of the process industry operations requires that we actively extract, represent and manage the knowledge embedded in operator procedures. This research is aimed at achieving this broad goal of managing action knowledge embedded in the procedures. The service extraction approach developed in this paper is one element of this overall goal. We hope that the approach outlined will serve as the basis for further discussion and enhancement.

Acknowledgements. We acknowledge the grant to explore knowledge management concerns in petrochemical industry by the Center for Operator Performance (Center); the numerous conversations with several members of the Center; and the access to operators and procedures provided by member organizations of the Center.

References

1. Ham, D.-H., Yoon, W.C.: The Training Effects of Principle Knowledge on Fault Diagnosis Performance. Human Factors and Ergonomics in Manufacturing & Service Industries 17, 263–282 (2007)
2. Jain, A., Akkisetty, V.P.K., Hailemariam, L.M., Suresh, P., Joglekar, G., Venkatasubramanian, V., Morris, K.R., Reklaitis, G.V.: An Ontological Framework for Knowledge Modeling in Pharmaceutical Product Development. In: The Preliminary Program for 2007 Annual Meeting, New Brunswick, NJ (2007)

3. Jia, X., Zhang, Z., Tian, X.: Research and Application on Typical Process Knowledge Discovery in Mechanical Manufacturing Enterprise. In: First International Workshop on Knowledge Discovery and Data Mining, WKDD 2008, pp. 196–200 (2008)
4. Madsen, E.S., Riis, J.O., Waehrens, B.V.: The Knowledge Dimension of Manufacturing Transfers: A Method for Identifying Hidden Knowledge. Strategic Outsourcing: An International Journal 1, 198–209 (2008)
5. Ohashi, M., Yuki, Y.: Productivity Improvement by Automating Operators' Knowledge and Experience. In: Proceedings of the 41st SICE Annual Conference, vol. 2, pp. 999–1003 (2002)
6. Backus, B.A.: Factors Related to The Economic Sustainability of Two Year Chemistry-Based Technology Training Programs Education. Oregon State University Corvallis (2009)
7. Bera, P., Wand, Y.: A Framework to Clarify the Role of Knowledge Management Systems. In: Pacific Asia Conference on Information Systems, Hyderabad, India, pp. 68–81 (2009)
8. Wolf, F.G.: Operationalizing and Testing Normal Accident Theory in Petrochemical Plants and Refineries. Production and Operations Management 10, 292–305 (2001)
9. Grote, G.: Diagnosis of Safety Culture: A Replication and Extension towards Assessing "Safe" Organizational Change Processes. Safety Science 46, 450–460 (2008)
10. Hsua, S.H., Lee, C.-C., Wu, M.-C., Takano, K.: A Cross-Cultural Study of Organizational Factors on Safety: Japanese vs. Taiwanese Oil Refinery Plants. Accident Analysis & Prevention 40, 24–34 (2008)
11. Rowe, K.: Bottlenecks and Constraints within The Local Labour Market for Engineers in The Petrochemical Industry Sector: A Case Study of Engen Refinery, Wentworth. Industrial, Organisational and Labour Studies. University of KwaZulu-Natal, Durban (2009)
12. Nivolianitou, Z., Konstandinidou, M., Michalis, C.: Statistical Analysis of Major Accidents in Petrochemical Industry Notified to the Major Accident Reporting System (MARS). Journal of Hazardous Materials 137, 1–7 (2006)
13. Strahan, A.: Oil Industry in U.S. Needs Engineers, Courts Retirees, Students (2005), `http://www.bloomberg.com/apps/news?pid=newsarchive&sid=abwZ5OoPVmZE&refer=us#share`
14. Jamieson, G.A.: Empirical Evaluation of an Industrial Application of Ecological Interface Design. Human Factors and Ergonomics Society Annual Meeting Proceedings 46, 536–540 (2002)
15. Newell, A.: The Knowledge Level. Artificial Intelligence 18, 87–127 (1982)
16. Newell, A., Simon, H.A.: Computer Science as Empirical Inquiry: Symbols and Search. Communications of the ACM 19, 113–126 (1976)
17. Blosch, M.: Pragmatism and Organizational Knowledge Management. Knowledge and Process Management 8, 39–47 (2001)
18. Nonaka, I.: A Dynamic Theory of Organizational Knowledge Creation. Organization Science 5, 14–37 (1994)
19. Kannan, K., Srivastava, B.: Promoting Reuse via Extraction of Domain Concepts and Service Abstractions from Design Diagrams. In: IEEE International Conference on Services Computing, Hawaii, vol. 1, pp. 265–272 (2008)
20. Matos, C.: Service Extraction from Legacy Systems. In: Ehrig, H., Heckel, R., Rozenberg, G., Taentzer, G. (eds.) ICGT 2008. LNCS, vol. 5214, pp. 505–507. Springer, Heidelberg (2008)

21. Jansen, S.: ServiciFi: Service Extraction from Decomposed Software Monoliths in The Financial Domain (2010), `http://servicifi.files.wordpress.com/2010/06/serviceextraction.pdf`
22. Sirin, E., Parsia, B., Hendler, J.: Filtering and Selecting Semantic Web Services with Interactive Composition Techniques. IEEE Intelligent Systems 19, 42–49 (2004)
23. Pearl, J.: Heuristics: Intelligent Search Strategies for Computer Problem Solving. Addison-Wesley, Reading (1984)
24. Newman, A.J., Pancheva, R., Ozawa, K., Neville, H.J., Ullman, M.T.: An Event-Related fMRI Study of Syntactic and Semantic Violations. Journal of Psycholinguistic Research 30, 339–364 (2001)
25. Charniak, E.: Statistical Techniques for Natural Language Parsing. AI Magazine 18, 33–43 (1997)
26. Fellbaum, C.: WordNet an Electronic Lexical Database. MIT Press, Cambridge (1999)
27. Dittrich, K., Dayal, U., Buchmann, A., McCarthy, D.: Rules Are Objects Too: A Knowledge Model for an Active, Object-Oriented Database System. In: Dittrich, K.R. (ed.) OODBS 1988. LNCS, vol. 334, pp. 129–143. Springer, Heidelberg (1988)
28. Scriven, M.: Types of Evaluation and Types of Evaluator. American Journal of Evaluation 17, 151–161 (1996)
29. Cleven, A., Gubler, P., Kai, M.H.: Design Alternatives for the Evaluation of Design Science Research Artifacts. In: Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology, Philadelphia, Pennsylvania (2009)
30. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28, 75–105 (2004)
31. Vaishnavi, V.K., Buchanan, G.C., Nevins, A.J.: Smart Objects: A Tool for Building Intelligent Support Systems. In: Proceeding of the Twenty-Sixth Hawaii International Conference on System Sciences, Wailea, HI, vol. 3, pp. 93–102 (1993)
32. Lee, J.-S., Hsu, P.-L.: Remote Supervisory Control of the Human-in-The-Loop System by Using Petri Nets and Java. IEEE Transactions on Industrial Electronics 50, 431–439 (2003)
33. Cranor, L.F.: A Framework for Reasoning about the Human in The Loop. In: Proceedings of the 1st Conference on Usability, Psychology, and Security, San Francisco, California (2008)

Reconsidering Modular Design Rules in a Dynamic Service Context

Jason Nichols[1], Michael Goul[2], Kevin Dooley[2], and Haluk Demirkan[2]

[1] Spears School of Business, Oklahoma State University
[2] W.P. Carey School of Business, Arizona State University

Abstract. Modular design rules are rooted in a tradition of process design for physical production. In response to an emerging information systems research agenda for design logic in the realm of services and digital goods, and through the lens of dynamic capabilities theory, the research presented here re-examines traditional modular design in the context of a service-centric volatile marketplace. A complex adaptive systems simulation artifact from prior literature is augmented with a novel operationalization of market volatility, and a series of hypotheses are tested that demonstrate a need for revision of modular design rules in a dynamic context. Rules that have historically isolated the modular design decision to characterizations of task interaction are expanded to incorporate a new objective: adaptive parity with the environment. It is the goal of this continuing research stream to make early contributions in the recently proposed agenda for new organizing logic in digital innovation and services.

Keywords: modular design, service design rules, dynamic capabilities theory, complex adaptive systems, simulation, service-oriented enterprise.

1 Introduction

In a recent article, Yoo et al. propose an information systems research agenda in support of an emerging "layered modular architecture" for firms, that fuses technology with digital production and embedded services (Yoo, Henfridsson, & Lyytinen, 2010). This new agenda re-contextualizes modular design principles within the world of digital production and services science. The maturing notion of the Service-Oriented Enterprise (SOE) can be one avenue through which information systems are brought to bear within this context (Cherbakov, Galambos, Harishankar, Kalyana, & Rackham, 2005). However, prior testing and validation of long-standing modular design principles, a critical foundation upon which SOE technologies can be used to design processes (choreographies in SOE parlance), lack a characterization of the volatile environment within which the designs execute. These design rules are historically task-interaction-centric. It is possible that the very design principles, which dictate service and digital product configuration, must be adjusted to account for this new context. The research stream depicted in this manuscript seeks to make early headway within Yoo et al.'s agenda for layered modular architecture by 1) contextualizing volatility in the service and digital production landscape through the lens of dynamic capabilities theory, 2) expanding a well-recognized modular design

H. Jain, A.P. Sinha, and P. Vitharana (Eds.): DESRIST 2011, LNCS 6629, pp. 350–365, 2011.

simulation artifact to incorporate an operationalization of the market volatility that is characteristic of the services context, and 3) presenting theoretically grounded revisions to classic modular design rules based on the results of hypothesis testing within the revised artifact.

With a transition from an industrial to a hybrid service/industrial or perhaps increasingly pure service-based economy, the lens through which researchers examine competition between firms has evolved to emphasize the ability of the firm to adapt and respond to increasingly regular shifts in market preferences. Dynamic capabilities theory represents one such perspective, contextualizing market shifts as disruptions and a firm's competitiveness as the ability to adequately respond by reorganizing and/or refactoring capabilities in order to continue to provide value within the context of the market's new preference landscape (Eisenhardt & Martin, 2000; Helfat & Peteraf, 2003; Makadok, 2001; Pavlou & El Sawy, 2005; Teece, Pisano, & Shuen, 1997; Winter, 2003). Figure 1 depicts the lifecycle of a firm's capabilities as the firm faces shifting market dynamics, leading to disruption.

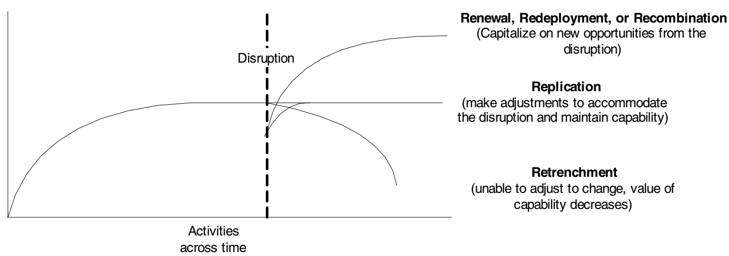


Fig. 1. A capability's lifecycle and the role of disruption (Helfat & Peteraf, 2003)

Through the formalism of Dynamic Capabilities theory, key assertions regarding the role of information systems technology and design rules for process and service configuration become clear. Figure 2 positions these elements within the capabilities hierarchy, exposing their critical nature for the firm looking to compete in a service-driven economy, defined by their ability to respond affectively to market disruption.

Information systems technology is employed at the higher levels of the capabilities hierarchy, and provides a vehicle through which emerging characteristics of the market environment and process/service execution can be captured and monitored. The Service-Oriented Enterprise (SOE) represents such an organizational model, where information systems are employed to capture data from dynamic process executions (known as choreographies), as well as to sense and capture contextual information related to the environment within which these choreographies execute. One outcome of this model for information systems embedded within organizational processes is a critical data source that can be used to help guide future process and service design decisions.

Patterns in this emergent data can then be exposed to the design rules that the firm employs when constructing their customer-facing products and services. The products and services themselves are a result of the firm's "shop-floor" processes, which represent a firm's lowest and most volatile level within the capabilities hierarchy.

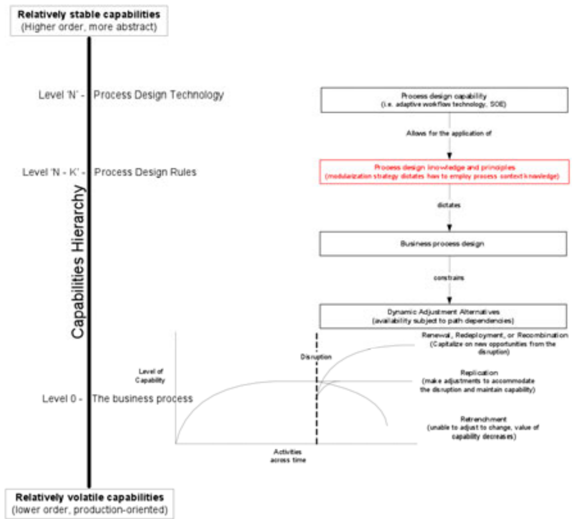


Fig. 2. The role of information systems and design rules in the capabilities hierarchy

Dynamic Capabilities theory asserts that design decisions made at the higher levels of the hierarchy ultimately dictate both performance and adaptability for the resultant products and services. These objectives (peak performance and adaptability) exist in conflict within themselves. Inherently, the best performing product or service offering is one that is completely tailored to market preferences at a current point in time. However, tailoring the production process of a product or service to perfectly match the current state of the market reduces the ability for that process to evolve at an increasing rate of change for market preferences. The tension between these two objectives is amplified within a service context, where firms compete by adapting to customer demand and customer demand evolves rapidly.

Within this context, the design rules that dictate the structure of lower level processes, by necessarily committing the process to a position along the spectrum of the performance/adaptability tradeoff, become critical to the success of the firm. Here, modular design has emerged as a key set of principles that support the dynamic capabilities notion of "planned adaptation" (Eisenhardt & Sull, 2001; Simon, 1993; Teece, et al., 1997). Research in this area began with the exploration of alternative configurations for complex systems (Simon, 1962). It was noted that hierarchical and decomposable systems demonstrated greater stability, and the research in this area evolved into the development and validation of tools for system decomposition. One such tool, the design structure matrix (DSM), is used to identify interactions between tasks in a workflow and then modularly cluster tasks so that the task groupings exhibit high intra, yet low inter-group interaction (Baldwin & Clark, 2000). Figure 3 represents a sampling of DSM across a number of task interaction scenarios.

Within a DSM each task is represented along the vertical and horizontal axes, and an "x" is placed at the intersection of a pair of tasks if those two tasks are dependent upon each other. The purpose of the DSM is to then reorganize the tasks in question

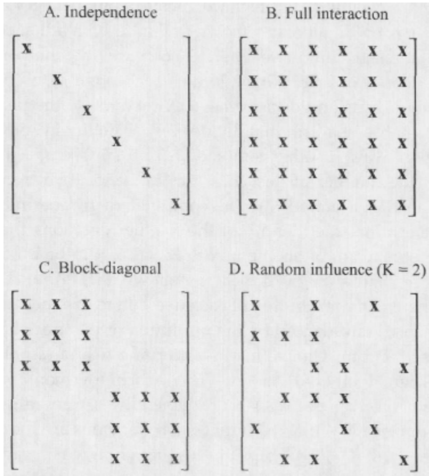


Fig. 3. DSM representing the interaction structure of a diverse set of patterns

so that they cluster together with minimal interaction across the clusters. The emergent nature of knowledge regarding task interaction and environment, and characterization of the marketplace as a rough landscape to which firms are regularly adapting products and services in order to meet demand, make complex adaptive systems simulation a natural test bed within which DSM-driven modular designs can be tested and the DSM tool examined for validity (Kauffman, 1993). Because of this, Kauffman's NK model has been used as the basis for a stream of research seeking to explore the efficacy of modular design based on task interaction. The results of these efforts have supported DSM-style modular design as the dominant modularization strategy, and a critical design principle has emerged from this research: when in doubt, under-modularize (Ethiraj & Levinthal, 2004b).

Ironically, however, the very nature of the evolving marketplace has been assumed away in these prior experiments. Specifically, performance landscapes for choreographies in the simulation trials have historically remained static. This is in stark contrast to the competitive conditions of a service-based context where customer preferences are fickle and rapidly changing. The research herein seeks to build off of existing work in the area by implementing a grounded characterization of environmental volatility within the same NK simulation model. DSM decomposition will be used as a baseline for modular choreography design, and performances for configuration adjustments from this baseline will be observed in order to explore possible adjustments to modular design principles in the face of a dynamic environment.

2 Hypotheses

Design research on modularization has always been about identifying a target characterization for hierarchical task decomposition. Prior research has led to a characterization emphasizing task interaction, and recommending decomposition so that inter-module interaction is minimized. This DSM-style decomposition represents an effort to balance the adaptability of increased modularization with the peak performances of more tightly coupled designs. In this exploratory study, aspects of the environment, specifically characteristics of market volatility, are introduced. Therefore, hypotheses to identify adjustments that may need to be made to target decomposition (service abstraction) guidelines in the face of varying environmental conditions are organized under a guiding proposition that attempts to capture the inherent tension between modularization for adaptability while maintaining performance levels in the face of volatility. Table 1 depicts the "adaptive parity" proposition and related hypotheses. Adaptive parity simply suggests that a modular design's adaptability must necessarily keep up with the rate at which disruption occurs within the execution context. This proposition suggests that, in a volatile market, designs should err on the side of increased modularity. Adaptive parity is a marked revision to existing design rules that propose under-modularization in the case of design uncertainty (Ethiraj & Levinthal, 2004b). For each experiment, classic DSM decomposition represents the "true" underlying task interaction structure; the baseline off of which adjustments are made to test performance impacts under varying environmental contexts.

Table 1. The adaptive parity proposition and related hypotheses

***Adaptive parity proposition*: the target level of modularization in a volatile environment becomes the level at which organizational adaptability is in parity with environmental volatility.**
Hypothesis 1 *(impact of volatility hypothesis)*: As volatility increases for a choreography with a level of service abstraction set to mirror that of the "true" modular structure, the performance of the choreography will degrade.
Hypothesis 2 *(adaptation hypothesis)*: In a volatile environment, a choreography with an increased level of service abstraction will outperform a choreography that has a service abstraction level at or below that of the "true" modular structure.
Hypothesis 3 *(below parity hypothesis)*: Additional levels of service abstraction that serve to bring the adaptability of the choreography closer to parity with the volatility of the environment will improve choreography performance.
Hypothesis 4 *(above parity hypothesis)*: Increasing the level of service abstraction beyond that which balances adaptability with volatility will degrade the performance of a choreography.
Hypothesis 5 *(decreasing returns to approaching parity hypothesis)*: The magnitude of performance improvement for adding an additional layer of service abstraction when approaching parity with the volatility of the environment monotonically decreases.

3 Methodology

The basis for this simulation model is adapted from Ethiraj (Ethiraj & Levinthal, 2004a, 2004b) who in turn credits Simon (1962) for the conceptual criterion used to convert Kauffaman's (1993) NK model with random interaction structure into a non-random interaction environment representing modules with high intra-module interaction and low inter-module interaction. The model begins with a set of thirty tasks, each capable of being configured in one of two alternative fashions (binary task variables). The tasks represent individual activities that need to be performed in a choreography in a Service Oriented Enterprise. Since each task in the choreography can take on one of two configurations, there are 2^{30} unique possible configurations for the set of tasks in the choreography. The performance of each task is a function of not only the configuration of the individual task, but the configuration of the tasks with which it interacts. This is in line with the common conceptualization of the activities within a firm occurring on a rugged and nonlinear landscape where unknown task interactions and explicit task structuring serve to increase the complexity of searching the landscape for performance peaks through local task performance improvement (Levinthal, 1997).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1		x	x	x	x	x																								
2	x		x	x	x	x																								
3	x	x		x	x	x																								
4	x	x	x		x	x																								
5	x	x	x	x		x																								
6	x	x	x	x	x																									
7						x		x	x	x	x	x																		
8							x		x	x	x	x																		
9							x	x		x	x	x																		
10							x	x	x		x	x																		
11							x	x	x	x		x																		
12							x	x	x	x	x																			
13												x		x	x	x	x	x												
14													x		x	x	x	x												
15													x	x		x	x	x												
16													x	x	x		x	x												
17													x	x	x	x		x												
18													x	x	x	x	x													
19																		x		x	x	x	x	x						
20																			x		x	x	x	x						
21																			x	x		x	x	x						
22																			x	x	x		x	x						
23																			x	x	x	x		x						
24																			x	x	x	x	x							
25																								x		x	x	x	x	x
26																									x		x	x	x	x
27																									x	x		x	x	x
28																									x	x	x		x	x
29																									x	x	x	x		x
30																									x	x	x	x	x	

Fig. 4. Interaction Structure: N=30, M=5

The interaction structure for the tasks in the choreography is presented in figure 4. Validation of the simulation instrument is a critical first step for this research. In order to achieve this through input/output comparison with the Ethiraj (2004b) study, the predominant interaction structure reported therein where N (the number of tasks) equals 30 and M (the number of modules) equals 5 is adopted. An “x” in a cell of the interaction matrix represents a relationship between the tasks designated by the index of the row and the index of the column where the intersection occurs. Therefore, the interaction matrix dictates the set of tasks whose configuration any individual task’s performance depends upon (along with its own configuration).

For example, based on the matrix in figure 9, task 5 (represented by the fifth row in the matrix) is dependent upon itself as well as tasks 1, 2, 3, 4, and 6 (as can be seen by the “x” in each of the respective columns in the matrix). Since each task can take on one of two configurations and the performance of task 5 is a function of six tasks (itself and five other tasks with which it interacts), there are 2^6 or 64 possible performance values for task 5. Task 7, however, is dependent upon itself as well as tasks 8, 9, 10, 11, and 12 (representing the within-module high degree of task interaction) and also task 6 (representing the weaker “interface” interaction between modules (Simon, 1962)). Therefore task 7 can take on 2^7 or 128 possible performance values. To operationalize performance, a value for each possible configuration of a task and the tasks with which it interacts is randomly drawn from the uniform distribution U[0, 1]. The overall performance of the choreography is then computed as the simple average of all task performances.

When the simulation is started, each period every service entity (or module) in the choreography is allowed to search for a local adaptation to improve performance. The service entity is only able to observe the performance of the tasks within the entity itself. The performance of the service entity is calculated as the average of the performance of the three tasks it contains. The entity calculates its performance, randomly selects one of its tasks, reconfigures the task, and compares its overall local performance with the reconfigured task to that of the overall local performance before the change. If the local performance of the service entity improves, then the adaptation is kept. Otherwise, it is discarded. Each service entity in the choreography is allowed to search for one local adaptation per period. Through this process, the central tension between increased perception of the impact of a local adaptation and the speed at which the choreography can evolve is enacted. An over-modularized choreography design ($M>5$) is allowed extra local adaptations per period, but the adaptations ignore a higher degree of task interactions when they are put into place (potentially degrading the performance of other tasks through their adoption). An under-modularized choreography design ($M<5$) has greater perspective on the relationship between the tasks in the choreography, however is considerably slower to evolve due to fewer opportunities for local adaptation per period. Validation of the simulation model occurs through an input/output comparison of the patterns of performance emergent in both this study's implementation of the model and the implementation in the Ethiraj (2004b) study. The pattern of concern involves the relative performance trends of local adaptation for choreographies of size $N=30$ where one is over-modularized ($M=10$), one is under-modularized ($M=3$), and one is modularized in congruence with the true underlying interaction structure ($M=5$).

3.1 Modeling Environmental Volatility

The NK interaction model is then augmented through the inclusion of theoretically grounded operationalizations of marketplace volatility. Volatility in the marketplace is characterized by how often a disruption occurs, and how large the disruption is (Jurkovich, 1974; Miles, Snow, & Pfeffer, 1974). This study adopts a four point scale for marketplace volatility ranging from [0, 3]. Volatility is operationalized in the performance landscape underlying the NK interaction model through a re-computation of the performance landscape for tasks. The four point scale characterizes the rate of re-computation and the number of tasks being recomputed (the magnitude of the re-computation). So, for example, a volatility level of 0 emulates the static performance landscape of prior studies where each period there is a 0% chance that any task has its performance landscape recomputed. A volatility level of 1 represents a 25% chance that 25% of the tasks will have their performance landscape re-computed. A level of 2 represents a 50% chance that 50% of the tasks are recomputed, and so on. If the performance landscape for a task is conceptualized as a series of peaks and valleys, then the re-computation of a task's performance landscape is akin to rearranging the peaks and valleys that the firm navigates while searching for better task configurations to achieve a higher performance peak. Implementing market volatility in this fashion emulates the evolving needs of the marketplace that change the requirements for services provided by the choreographies of a Service Oriented Enterprise.

3.2 Simulation Validation

The NK interaction model underlying the simulation for this research has been adapted from the Ethiraj (2004b) study, specifically targeting his work on local adaptation for varying degrees of uniform modularity. In order to validate the comparability of the current study's model with the Ethiraj model, the pattern of performance for equivalent model inputs should be consistent across the two implementations (O'Leary, Goul, Moffit, & Radwan, 1990). Figure 5 depicts the output from the Ethiraj implementation for a scenario where N (the number of tasks) equals 30 and M (the number of modules in the true underlying interaction matrix) equals 5. For comparison, the validation run for the current study's model (presented in figure 6) includes three characteristic modularization levels: Under-modularization (3 modules), over-modularization (10 modules), and "true" modularization (5 modules). The pattern, as observed in Ethiraj (2004b) includes the following characteristics:

- "true" modularization creates a performance upper-bound for alternative modularization strategies.
- Under-modularization approaches the performance level of "true" modularization, but does so at a slower pace.
- Over-modularization approaches an asymptote below the performance level of "true" modularization.
- Initially, over-modularization out-performs under-modularization before under-modularization overtakes it while approaching a higher performance asymptote at a slower pace.

The same pattern for performance outcomes of the different modularization strategies can be observed in the validation output for the current implementation. Initially, over-modularization outpaces under-modularization because of its increased adaptability. However, under-modularization catches up and overtakes it on its way

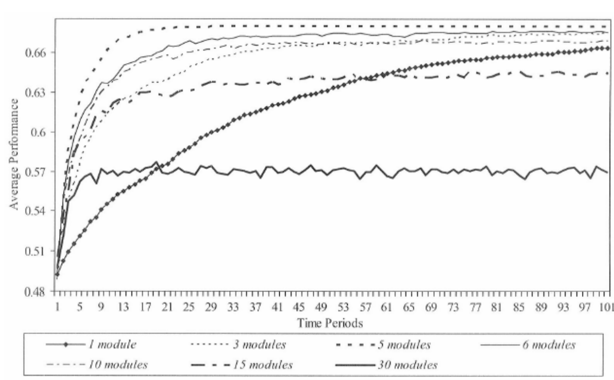


Fig. 5. Ethiraj output for N=30, M=5

to a higher performance asymptote at the level of the “true” modularization strategy. A comparison of the trends in output across figures 5 and 6 demonstrate that the current implementation captures the behavior of Ethiraj’s prior implementation and replicates the pattern of results reported in his study. The baseline environment for the current implementation (volatility and engagement complexity set to 0) is congruent with the static performance landscape examined in prior studies.

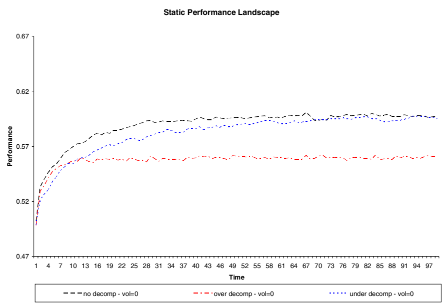


Fig. 6. Validation output: N=30, M=5

4 Hypothesis Testing and Results

The results of hypothesis testing are presented in bar charts such as that depicted below.

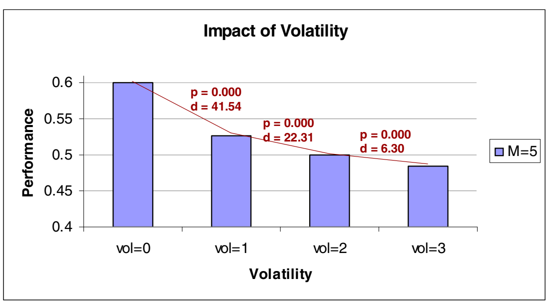


Fig. 7. Example outcome from hypothesis testing

This example chart is from the results of testing hypothesis 1. Each bar represents the average performance, across 1000 independent simulation runs, for a specific environmental and choreography configuration. For each hypothesis test the simulation runs were allowed to continue until performance stabilized (Jacklin, et al., 2005; Phattanasri & Loparo, 2005). The stable zone for each test spanned periods 70 through 100, therefore the height of each bar is calculated as the average of these thirty periods across the 1000 simulation runs. Because this study focuses on evaluating the impact of changes in environment and choreography configuration, Cohen's d was utilized as a measure of effect size across average performances as well as traditional tests for statistically significant differences. Both of these tests are reported in each chart, where the p-value represents statistical significance and the d value represents the magnitude of effect for a change from one configuration to the next. An arc is drawn from one bar to the next in order to illustrate the configurations being compared, and each arc is complimented with the results of both significance and effect size testing.

Performance, the dependent variable, is always represented as the Y-axis in the charts. The X-axis varies based on the specific independent variable being tested for each hypothesis. In the example above, the X-axis represents different levels of volatility. The complete set of variables manipulated across hypothesis testing include environmental volatility (represented in the charts as "vol" or "v"), the number of modules in the choreography ("m"), the choreography's engagement complexity ("ec"), and the total number of tasks in the choreography ("n"). For each test, the underlying interaction structure for the choreography is such that there exists 5 "true" modules (sets of tasks within which interaction is high, and across which interaction is low).

4.1 Analysis of Hypothesis 1

Hypothesis 1 (the impact of volatility hypothesis) states that the performance of a choreography designed through existing modular guidelines will degrade as the volatility of the environment increases.

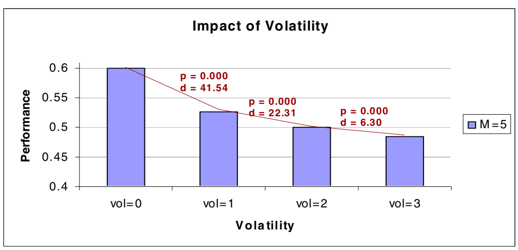


Fig. 8. Results of testing hypothesis 1

Prior modular guidelines recommend a structure that mirrors the "true" task interaction structure of the choreography. This is to say that a properly designed choreography will be composed of sets of tasks where task interaction is contained within the sets and minimized across sets. For each configuration, represented as a bar in the chart below, the choreography was decomposed to mirror the 5 module design of the choreography's underlying task interaction structure. The X-axis represents increasing volatility. For each increase in environmental volatility, the degradation of performance is both statistically significant (p=0.000, alpha=.05) and of a large effect size (d>.8). Hypothesis 1 cannot be rejected, and a significant reduction in performance for a choreography designed through traditional modular guidelines is observed as environmental volatility increases.

4.2 Analysis of Hypothesis 2

Hypothesis 2 (the adaptation hypothesis) states that, in a volatile environment, a choreography designed with a higher degree of modularity will outperform those that are designed with a degree of modularity at or below that which is recommended by traditional modular design rules. The X-axis in the chart below represents granularity, where the three levels inspected are modular designs of 3 sets of tasks (under-modularized), 5 sets of tasks (the recommended configuration from traditional design guidelines), and 10 sets of tasks (representing a choreography with an increased degree of modularization). For each configuration, the environmental volatility was set to high (v=3).

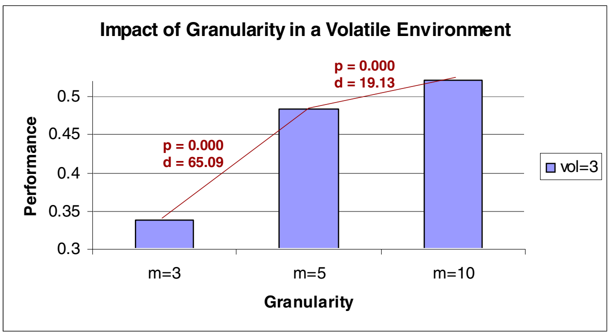


Fig. 9. Results of testing hypothesis 2

The choreography with a degree of modularity above that which is traditionally recommended outperformed both the “under-modularized” and the “right-sized” configurations in a volatile environmental context. The performance improvement above the traditionally designed choreography was both statistically significant (p=.000, alpha=.05) and of a large effect size (d>.08). Therefore, hypothesis 2 cannot be rejected. A significant improvement was observed in increasing the modularity of a choreography beyond that which is recommended under traditional modular design rules when facing a volatile environmental context.

4.3 Analysis of Hypothesis 3

Hypothesis 3 (the below parity hypothesis) states that increasing the modularity of a choreography in order to bring its adaptability closer to parity with the volatility of the environment will result in additional gains in performance. In order to test this hypothesis, the volatility of the environment is set to high (vol=3). In this context, there is a 75% chance each period that 75% of the task configurations in the underlying structure will be changed.

It should be noted here that the configuration of the tasks in the underlying structure represent the “goal” configuration for the choreographies in the simulation in order to achieve maximum performance. In this sense, the underlying task configuration represents market requirements for the choreography being evaluated.

As each choreography contains 30 tasks, this means that each period approximately 22 of the tasks will require reconfiguration in order to keep up with the degree of change occurring in requirements for the choreography. In the chart below, granularity is again represented across the X-axis.

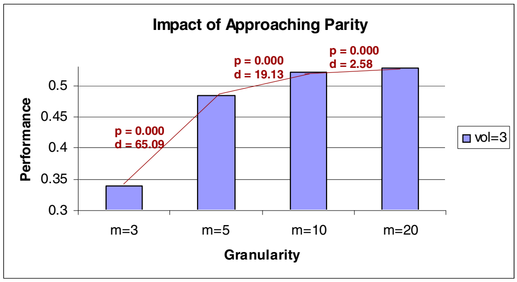


Fig. 10. Results of testing hypothesis 3

The degree of modularity is manipulated to span an under-modularized configuration (m=3), a configuration adhering to traditional modular design rules (m=5), an over-modularized configuration (m=10), and an additional over-modularized configuration close to but just below parity with the volatility of the environment (m=20). Each period, every module is allowed to evaluate the performance improvement local to that module for implementing one adaptation to one task within that module. In the m=20 configuration, therefore, 20 adaptations are possible per period. This is close to but just below the 22 requirements adjustments possible each period in the underlying choreography structure. In order to test hypothesis 3, the comparison of importance is that between the two over-modularized configurations of m=10 and m=20. Hypothesis 2 demonstrated a performance improvement for an over-modularized configuration in a volatile context, whereas hypothesis 3 is designed to examine the impact of additional modularization approaching parity with the environment. The performance improvement across these two configurations is both significant (p=0.000) and of a high effect size (d>.08). Hypothesis 3 cannot be rejected, and increases in modularity for a choreography that approach parity with the volatility of the environment result in additional performance improvements.

4.4 Analysis of Hypothesis 4

Hypothesis 4 states that modularizing past the point of parity with the environment will result in performance degradation. In order to test this, the same experimental setting from hypothesis 3 was used with the inclusion of a choreography configured with 25 modules (m=25). The adaptability of this configuration is above the degree of volatility represented in the environment (potential for 22 requirements adjustments per period).

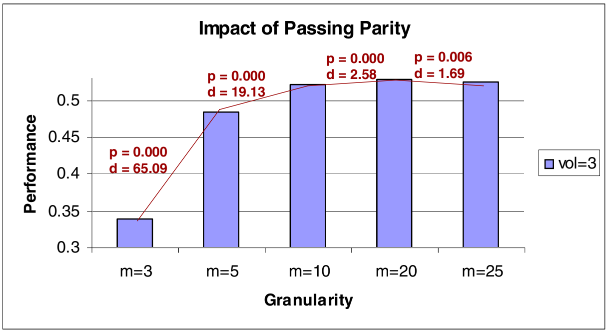


Fig. 11. Results of testing hypothesis 4

The comparison of interest for testing hypothesis 4 is that between the configuration where adaptability is just below parity (m=20) and the configuration where adaptability is above parity (m=25). A decrease in performance is observed after passing parity with the environment (p=0.006, alpha=.05) with a large effect size (d>.08). Hypothesis 4 cannot be rejected, and a significant performance reduction is observed when the adaptability of a choreography is increased past that which is needed to keep up with the volatility of the environment.

4.5 Analysis of Hypothesis 5

Hypothesis 5 (the decreasing returns to approaching parity hypothesis) states that the performance gains for increases in granularity of equal magnitudes will diminish. Put another way, the marginal performance gains for increased adaptability approaching the degree of volatility in the environment decrease.

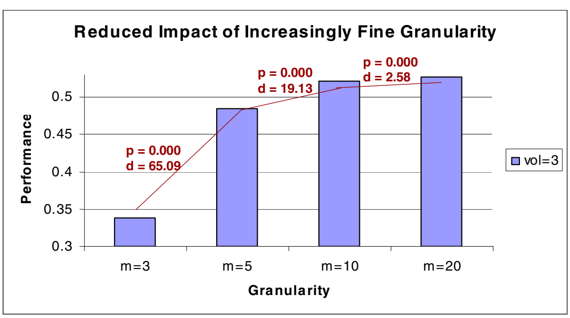


Fig. 12. Results of testing hypothesis 5

Hypothesis 3 demonstrated the significance of performance improvements as the modularity of the choreography increased from m=5 to m=10 and from m=10 to m=20. Hypothesis 5 is concerned with the magnitude of these improvements. Both increases in modularity, from m=5 to m=10 and from m=10 to m=20, represent a doubling of the degree of modularity for the choreography. However, the increase from m=5 to m=10 demonstrates a very large effect size (d=19.3) whereas the increase from m=10 to m=20 demonstrates a relatively much smaller effect size (d=2.58). Therefore, hypothesis 5 cannot be rejected; increases in modularity of equal magnitude when approaching parity with the volatility of the environment result in decreasing performance gains as the choreography becomes more modular.

5 Conclusion

DSM is a powerful tool for modularization in the context of the service-oriented enterprise. However, the injection of service management and design technologies avail the firm to a wealth of information regarding the context within which service choreographies are executed. This information is underutilized in DSM design, as the DSM tool itself focuses strictly on task interaction. The hypotheses tested here demonstrate that, in the face of environmental volatility, the DSM-oriented goal of minimizing inter-module interaction underperforms when compared to designs that target an alignment between design adaptability and the degree of volatility in the execution context; designs that approach adaptive parity with the environment. This shortcoming in DSM design is represented through the lens of dynamic capabilities theory in the figure below.

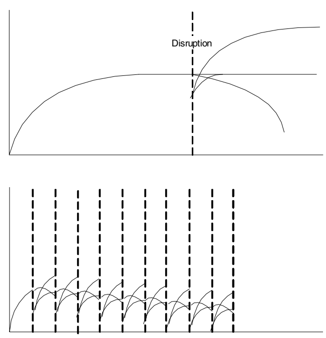


Fig. 13. Disruption limiting the expression of a firm's adaptive capabilities

If a choreography's design is not informed by characterizations of the volatility of the environment, adaptations to disruption are unable to adequately reflect performance gains before additional disruption occurs. Further, first steps towards approaching parity with the environment exhibit the greatest performance gains. Firms may then benefit from "convenience" opportunities for design adjustment rather than strict adherence to the emergent target degree of decomposition as they examine their context. This is in stark contrast to the previously established design principle, "when in doubt, undermodularize". The research here suggests that, in a volatile context, early notions of task interaction structure should lead to designs that err on the side of increased modularization. This fits nicely within Yoo et al.'s notion of design for "generativity", where components in a layered modular architecture are product agnostic (Yoo, et al., 2010). The volatility of the environment is such that future use for components cannot be known, thus driving increased abstraction into finer and more loosely coupled modular designs engineered for future reconfiguration.

The inclusion of environmental volatility in design research for modular systems is one step toward evolving modular design rules from their origins in the 1960's to a more contemporary perspective that captures the dynamism of competing in a service-driven context. From the results of this study, it is proposed that classic modular design mechanisms such as DSM, which rely on task interaction to dictate design, now form a foundation upon which aspects of the execution environment must come into play. Continuing research embeds further aspects of the environment into the simulation test bed, including the notion of engagement complexity from the coordination theory literature, and examines the interaction between competing environmental forces. Further, continuing work observes performance impacts at the level of the value chain in a service market, where each service entity is afforded their own distinct environmental profile. These efforts seek to contribute to the growing stream of information systems research that examines organizing logic and design rules at the intersection of technology as an organizational platform (e.g. SOE), and the growing marriage of digital products and services in the contemporary marketplace.

References

Baldwin, C., Clark, K.: Design Rules: The Power of Modularity. MIT Press, Cambridge (2000)

Cherbakov, L., Galambos, G., Harishankar, R., Kalyana, S., Rackham, G.: Impact of Service Orientation at the Business Level. IBM Systems Journal 44(4) (2005)

Eisenhardt, K., Martin, J.: Dynamic Capabilities: What Are They? Strategic Management Journal 21(10/11), 1105 (2000)

Eisenhardt, K., Sull, D.: Strategy as Simple Rules. Harvard Business Review 79(1), 106 (2001)

Ethiraj, S., Levinthal, D.: Bounded Rationality and the Search for Organizational Architecture: An Evolutionary Perspective on the Design of Organizations and Their Evolvability. Administrative Science Quarterly 49, 404 (2004a)

Ethiraj, S., Levinthal, D.: Modularity and Innovation in Complex Systems. Management Science 50(2), 159 (2004b)

Helfat, C., Peteraf, M.: The Dynamic Resource-Based View: Capability Lifecycles. Strategic Management Journal 24(10), 997 (2003)

Jacklin, S., Schumann, J., Pramod, G., Richard, M., Guenther, K., Soares, F.: Development of Advanced Verification and Validation Procedures and Tools for the Certification of Learning Systems in Aerospace Applications. Paper presented at the Infotech@Aerospace Conference, Arlington, Virginia (2005)

Jurkovich, R.: A Core Typology of Organizational Environments. Administrative Science Quarterly 19, 380 (1974)

Kauffman, S.: The Orgins of Order: Self-Organization and Selection in Evolution. Oxford University Press, New York (1993)

Levinthal, D.: Adaptation on Rugged Landscapes. Management Science 43(7), 934 (1997)

Makadok, R.: Toward a Synthesis of the Resource-Based and Dynamic-Capability Views of Rent Creation. Strategic Management Journal 22(5), 487 (2001)

Miles, R., Snow, C., Pfeffer, J.: Organization-Environment: Concepts and Issues. Industrial Relations 13, 244 (1974)

O'Leary, T., Goul, M., Moffit, K., Radwan, A.: Validating Expert Systems. IEEE Expert 51 (1990)

Pavlou, P., El Sawy, O.: Understanding the 'Black Box' of Dynamic Capabilities: A Missing Link to the Strategic Role of IT in Turbulent Environments. Under 3rd Round of Review - Management Science (2005)

Phattanasri, P., Loparo, K.: Verification and Validation of Complex Adaptive Systems. Paper Presented at the IJCNN 2005 - Verification, Validation, and Certification of Neuro-Adaptive Controllers in Safety-Related Areas, Montreal, Canada (2005)

Simon, H.: The Architecture of Complexity. Proc. Amer. Philos. Soc. 106, 467 (1962)

Simon, H.: Strategy and Organizational Evolution. Strategic Management Journal 14, 131 (1993)

Teece, D., Pisano, G., Shuen, A.: Dynamic Capabilities and Strategic Management. Strategic Management Journal 18(7), 509 (1997)

Winter, S.: Understanding Dynamic Capabilities. Strategic Management Journal 24(10), 991 (2003)

Yoo, Y., Henfridsson, O., Lyytinen, K.: The New Organizing Logic of Digital Innovation: An Agenda for Information Systems Research. Information Systems Research 21(4), 724–735 (2010)

Design Science in Service Research: A Framework-Based Review of IT Artifacts in Germany

Jörg Becker[1], Daniel Beverungen[1], Martin Matzner[1], Oliver Müller[2], and Jens Pöppelbuß[1]

[1] University of Muenster, European Research Center for Information Systems (ERCIS), Leonardo-Campus 3, 48149 Münster, Germany
{joerg.becker,daniel.beverungen,martin.matzner, jens.poeppelbuss}@ercis.uni-muenster.de
[2] University of Liechtenstein, Institute of Information Systems, Fürst-Franz-Josef-Strasse 21, 9490 Vaduz, Liechtenstein,
oliver.mueller@uni.li

Abstract. The purpose of this study is to analyze the nature of IT artifacts that have been proposed in the emerging discipline of Service Science, Management and Engineering (SSME) as well as to provide further directions for design research in the service discipline. We review a sample of 123 service-related IT artifacts – that we identified on a German online research portal – by coding them with a framework for design research in the service science discipline. The key insights derived from the analysis are: (1) methods dominate other artifact types; (2) instantiations are almost exclusively developed for supporting the potential dimension of services; (3) research on customer solutions focuses on an inside-out perspective; (4) new constructs are predominantly developed for modeling the outcome dimension of services; (5) artifacts often possess a narrow scope; and (6) artifacts are seldom instantiated into software tools. These novel insights are expected to guide future design research in the service discipline by identifying areas which have only been sparsely addressed by design research or are yet to evolve to a sufficient state of maturity. Our approach is original as it features an early and innovative endeavor for identifying the nature of IT artifacts in SSME.

Keywords: Design Science, Service Science, IT Artifacts, Germany, Hybrid Value Creation, Customer Solutions, Product-Service Systems.

1 Introduction

Over the last decades, we have been witnessing a transition from a primarily goods-based to a more and more service-based economy in most developed countries [1]. Today, services are ubiquitous and they account, for example, for more than 80% of the gross domestic product (GDP) and total employment of the United States and about 70% of the GDP in Germany [2-4]. Interestingly, "[t]he service sector accounts for most of the world's economic activity, but it's the least-studied part of the economy" [5, p. 71]. Even up to today, researchers and practitioners within and across

H. Jain, A.P. Sinha, and P. Vitharana (Eds.): DESRIST 2011, LNCS 6629, pp. 366–375, 2011.

fields have not yet agreed upon a common definition of the term “service” [6]. Researchers from different disciplines have so far investigated the phenomenon from rather distinct angles, e.g., from an economic, business, or technical perspective [7].

Information Systems (IS) is an integrative research discipline “that is at the intersection of knowledge of the properties of physical objects (machines) and knowledge of human behavior” [8, p. 613]. Therefore, IS not only emphasizes to build theories that represent knowledge on how the world is like, but also focuses on engineering IT artifacts. Orlikowski and Iacono [9] call the IT artifact the “core subject matter” of IS research, while Gregor [8] states that what “distinguishes IS from other fields is that it concerns the use of artifacts in human-machine systems.” With appropriately designed IT artifacts, management and engineering problems in service systems can be supported, such as defining service portfolios, customizing individual value propositions for customers, and efficiently delivering services. Hence, the design and evaluation of IT artifacts for engineering and managing service systems is a major contribution IS research can render [10].

Against this background, the goal of this study is to review existing contributions of design science research to the service science discipline in Germany, which has not been in the main focus of similar studies [7,11]. We chose the field of hybrid customer solutions as an exemplary subset of the service discipline. Hybrid customer solutions (also referred to as hybrid products) are integrated bundles of physical goods and related value-added services that are intended to jointly solve a specific customer problem. Focusing on this research stream seems especially fruitful since numerous researchers, companies, and funding agencies have been involved in governmentally funded research projects in this area during the recent years. The purpose of this study is to provide these stakeholders with a status quo and perspectives for further design science research in the service discipline. Using an existing framework [12] as a device of mind, we analyzed a repository of 123 research results that had been published on the German online research portal “Research Map of Hybrid Value Creation” (German: Forschungslandkarte zur hybriden Wertschöpfung). This web portal collects, categorizes, and shares results of German service science research projects. The remainder of the paper is structured as follows: Next, we briefly review the framework for design science research in the service science discipline. With this framework, we then analyze artifacts that have been contributed to the research on hybrid customer solutions as a part of service science in Germany. Key insights are discussed subsequently. Finally, we give a brief conclusion and discuss limitations and prospects of the presented study.

2 Framework for Design Science Research in Service Science

In previous work [12], we developed a framework for structuring design-oriented research activities in the area of service management and engineering (Figure 1). The original framework encompasses three dimensions (artifact type; service perspective; and level of analysis, i.e., macro-, meso-, and micro-level), from which the following two form the basis for our analysis presented in this paper.

The first dimension is based on the outputs of design science research constituted by the four types of IT artifacts according to March and Smith [13]. *Constructs* form the vocabulary of a domain. They build the basis for defining problems and specifying their solutions. Modeling languages, such as the Business Process Model and Notation (BPMN) are common collections of constructs. *Models* are sets of statements expressing relationships between constructs. Reference models, for example, describe a class of real-world phenomena on an abstract level. Their purpose is to give guidance to the design of other company-specific models. *Methods* are sequences of steps used to perform a task. Typical examples are algorithms, procedures, or guidelines. Constructs and models typically represent the inputs and outputs of methods. Service Blueprinting [14], for instance, is a well-known exemplar of a method for service design. *Instantiations* are realizations of constructs, models, or methods in information systems. Instantiations are valuable for demonstrating the utility of artifacts. In addition, they test the feasibility of both the design process and the designed artifact. The utility and feasibility of a method for simulating service processes, for instance, can be demonstrated by a software tool.

The second dimension of our framework comprises different perspectives on the phenomenon of service [15]. The *potential* perspective accentuates that firms have to build up resources in order to provide services to clients. It focuses on the infrastructure basis which is used to design, configure, offer, and deliver services. Those resources might be operant resources (i.e., resources that operate on other resources, such as human resources, knowledge, or skills) or operand resources (i.e., resources to work on, such as raw material) [16]. The *process* perspective focuses on the business processes and activities for delivering services to customers. Due to the fact that value co-creation is a key characteristic of services [17], one central challenge in business process design is to determine the degrees of cooperation and visibility for each activity of the service process. The *outcome* perspective is concerned with determining the structure and the functional and non-functional properties of a service. In contrast to the process perspective, which focuses on how a service is delivered, it focuses on what is delivered. The *market* perspective respects the customer as a co-creator of value. The market perspective therefore comprises tasks such as identifying customer problems or determining a customer's willingness to pay for particular service offerings.

The two perspectives are in line with the components of other frameworks that have been developed to systematically study services or service systems respectively [18-21]. To illustrate the application of the proposed framework, Fig. 1 shows a classification of fictional artifacts. A typical artifact of the *construct-outcome* cell would be a modeling language that allows for an unambiguous specification of the function and form of a service. Similar modeling languages, e.g., EXPRESS-G / STEP, are commonly used to describe physical goods in a standardized and machine-readable format. However, due to the distinct characteristics of services widely accepted standards for the description of services are still missing. A typical artifact categorized in the *model-potential* cell would be a reference model depicting a best-practice organizational structure for service units in a certain industry. A mathematical procedure to calculate the optimal price of a service would be a compelling representative of the *method-market* cell. Such price optimizations for services are complex tasks due to the heterogeneous and perishable nature of services. An example for the *instantiation-process* cell would be a workflow system supporting the execution of service processes.

	Potential	Process	Outcome	Market
Construct	Modeling language to specify the resource infrastructure for hybrid customer solutions	Modeling language to specify service processes and manufacturing processes	Modeling language to specify the function and form of services	Modeling language to specify customer preferences and competitor prices
Model	Reference model for the organizational structure of service units	Conceptual framework for service operations	Models that specify the function and properties of services and physical goods building blocks	Recommendation model for calculating the willingness to pay of a customer
Method	Method for identifying, implementing, and utilizing KPIs for a productivity management for customer solutions	Method for configuring business processes	Method for configuring customer solutions from a predefined solution space	Mathematical procedure to calculate the optimal price of a service
Instantiation	Software tool for defining and compiling reports on capacity utilization and organizational learning	Workflow system supporting service delivery processes	Software workbench for defining and configuring hybrid customer solutions	Model-based recommender system for marketing hybrid customer solutions

Fig. 1. Framework for design science research in service science [12] and exemplary artifacts

3 Review of IT Artifacts

3.1 Data Collection

In the following, we apply the framework for analyzing the area of hybrid customer solutions in Germany, a specific research stream within service science. The objective is to analyze the research results that design-oriented research has so far contributed to this very subarea. A substantial collection of such research results is listed in the Research Map of Hybrid Value Creation, which is accessible at http://www.forschungslandkarte-hybridewertschoepfung.de. This web portal invites researchers and practitioners to publish, categorize, and share research results on the engineering and management of hybrid customer solutions. Up to today, more than 300 users have registered at the portal, which underlines the site's status as a valuable source of information. We were able to identify a total of 123 research results listed in the portal (as of 2010-11-29). Since we intended to analyze the contributions of design science research only, we excluded theories and purely empirical work (about 1/3 of all entries) from our further analysis. 78 entries remained for analysis.

3.2 Data Analysis

We conducted a structured content analysis and drew the relevant data from the descriptions of research results as published on the abovementioned online portal. In a content analysis, case descriptions and other accessible sources of data are coded. Multiple readings of data and multiple coders are employed to enhance reliability and validity of the analysis [22]. Content analysis may be used in an inductive or deductive way [23]. We pursued a deductive approach as the framework introduced in

Section 2 provided the categorization matrix for our structured content analysis. Each of the 78 descriptions of design-oriented research results was thoroughly studied and classified according to the framework's dimensions. We solely relied on the textual descriptions of the research results as a basis for coding and neglected the categorizations which the portal users had already assigned to their contributions. This was necessary, since the categorizations provided by portal users were partly incomplete or contradictory to the textual descriptions. The categorization of research results was done in a team of four researchers. For each research portal entry, two researchers independently coded the textual description. The four service perspectives and four artifact types together represent eight characteristics that were used to describe an artifact. It was allowed that a particular artifact may represent both more than one service perspective and more than one artifact type. After the independent categorization by two researchers, 11 out of the 78 results featured completely consistent classification across these eight characteristics. Inter-coder-reliability – computed based on a pairwise comparison of the coders' decisions made for each attribute – showed that in more than half of the decisions the reviewers came to the same result (Perc. Agreement: 55.6%, Scott's Pi: 0.051, Cohen's Kappa: 0.054, Krippendorff's Alpha (nominal): 0.051, N Agreements: 347, N Disagreements: 277, N Decisions: 1248) [24]. In all other cases, the remaining two researchers mutually analyzed the deviations and agreed on appropriate classifications.

3.3 Results

The analysis shows how often research results of a certain artifact type and a certain perspective on the service phenomenon have been entered into the research portal. Some artifacts are quite specialized, i.e., they represent a specific artifact type and are supposed to support a particular service perspective only. Other entries, however, are meant to address multiple perspectives and cannot clearly be assigned to one artifact type only. This is illustrated by the sums for the two analysis dimensions that both exceed the total number of artifacts (Table 1).

Table 1. Number of artifacts by service perspectives and artifact types

	Service perspective				Artifact type			
	Potential	Process	Outcome	Market	Construct	Model	Method	Instantiation
# Artifacts	45	34	35	29	26	22	47	20

With regard to artifact type, methods dominate by far. 47 out of the 78 artifacts are categorized as methods. Constructs (26), models (22) and instantiations (20) are found less often than methods. Concerning service perspectives, the analyzed artifacts predominantly address the potential perspective (45). The market perspective is addressed by 29 research results only.

We conducted a normalization of values by weighing the values according to the total number of classifications for each research result and each dimension of analysis. Thus, we made sure that all research results contribute equally to the further analysis. The cell values for each research entry were calculated as 'artifact type

Table 2. Cross tabulation of service perspectives and artifact types

	Potential	Process	Outcome	Market
Construct	13 (2.6)	14 (3.6)	16 (5.7)	9 (3.0)
Model	13 (5.4)	14 (5.0)	7 (1.9)	8 (2.9)
Method	29 (11.4)	21 (7.1)	21 (7.5)	17 (8.)
Instantiation	14 (6.1)	8 (2.8)	7 (2.69)	6 (2.2)

value' times 'perspective value'. For instance, the framework of hybrid value creation [25], which is a model, was found to address both potential and process dimension. It therefore contributes to the first two cells in the second line with a value of 0.5 each. Table 2 shows the values before and after (in brackets) normalization.

Looking at the combination of both classification aspects, the peak is at 29 artifacts that represent methods for the potential perspective. The normalized value for this perspective-artifact combination is 11.4. Instantiations for the market perspective are scarce (6 instances, normalized value of 2.2). The normalized value for models that address the outcome perspective (1.9) is even smaller. Instantiations, i.e. software tools, are almost exclusively used to support the potential perspective. 14 out of a total of 20 instantiations have this scope. Similarly, models are hardly found apart from the potential and process perspectives. Constructs are especially used to describe the outcome perspective of customer solutions (16 out of 26, normalized: 5.7). Fig. 2 charts the normalized results.

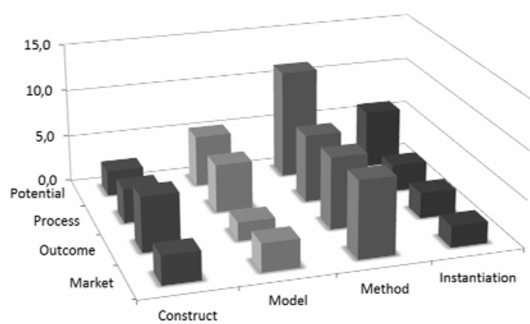


Fig. 2. Visualization of normalized results

Research results can be assigned to more than one characteristic in each dimension. Hence, it is promising to analyze, what combinations of characteristics within one dimension appear in conjunction with each other. As is illustrated in Table 3, by far the most frequent combination of two artifact types is the combination of constructs and methods (16 research results, i.e., 20.5% of all research results). At a closer look, this stems from modeling languages that comprise modeling constructs as well as they constitute methods for modeling. Other pairs are only seldom found or are non-existent. We further investigated whether three-out-of-four combinations or even combinations of all four artifact types can be found. Accordingly, we were able to identify 4 triples of models, methods, and instantiations and 2 triples of constructs, models, and instantiations. Surprisingly, although the construct-method combination is the most frequent pair of artifacts, no construct-method-instantiation triples have been found. Only one combination of all four artifact types was identified.

Table 3 shows the results of an analogous analysis of the service perspective dimension. Potential-process (13) and outcome-market (10) are the most frequent combinations. The occurrence of these combinations is quite self-evident, as potential and process are both rather inside-oriented and outcome and market more outside-oriented perspectives. By far the most numerous triple is the combination of the perspectives potential, process, and outcome (6). In total, 4 combinations of all four perspectives have been found.

Table 3. Absolute and relative frequencies of combinations of artifact types (left) and service perspectives (right)

	Construct	Model	Method	Instantiation	# (%)
1	●	○	○	○	5 (6.4%)
	○	●	○	○	11 (14.1%)
	○	○	●	○	23 (29.5%)
	○	○	○	●	10 (12.8%)
2	●	●	○	○	2 (2.6%)
	○	●	●	○	1 (1.3%)
	○	○	●	●	2 (2.6%)
	●	○	○	●	0 (0%)
	●	○	●	○	16 (20.5%)
	○	●	○	●	1 (1.3%)
3	●	●	●	○	0 (0%)
	○	●	●	●	4 (5.1%)
	●	●	○	●	2 (2.6%)
	●	○	●	●	0 (0%)
4	●	●	●	●	1 (1.3%)
					78 (100%)

	Potential	Process	Outcome	Market	# (%)
1	●	○	○	○	11 (14.1%)
	○	●	○	○	8 (10.3%)
	○	○	●	○	6 (7.7%)
	○	○	○	●	7 (9.0%)
2	●	●	○	○	13 (16.7%)
	○	●	●	○	0 (0%)
	○	○	●	●	10 (12.8%)
	●	○	○	●	3 (3.6%)
	●	○	●	○	5 (6.4%)
	○	●	○	●	0 (0%)
3	●	●	●	○	6 (7.7%)
	○	●	●	●	2 (2.6%)
	●	●	○	●	1 (1.3%)
	●	○	●	●	2 (2.6%)
4	●	●	●	●	4 (5.1%)
					78 (100%)

Finally, we analyzed whether there are many narrowly focused research results, solely concentrating on one characteristic in each dimension (see also Table 3). Among these focused research results, methods (23) dominate the artifact dimension. The distribution in the service perspective dimension is more uniform, with potential being the most frequent perspective.

4 Discussion

The objective of the research portal that we used as the data source for our analysis is to provide an overview of research that is already completed, still ongoing, or – indirectly – yet to be tackled in future research on hybrid costumer solutions.

Therefore, the results of our study can serve researchers in identifying hot spots and blind spots in this particular field. Some key insights, which we were able to derive from our study, are discussed in the following subsections.

Method is the dominating IT artifact type: Methods are the dominating type of artifact in research on hybrid customer solutions. Methods identified are, for instance, procedure models and guidelines that address tasks of all perspectives, but especially the potential perspective. The focus on methods can be explained by the research disciplines that are involved in service research in Germany and research on hybrid customer solutions in particular. Especially the German business administration and engineering disciplines both aim at proposing normative procedures that help to cope with economic and technological challenges. From the perspective of the involved researchers, methods obviously seem to provide a more innovative contribution to the body of knowledge than constructs or models do, which are in essence the inputs and outputs of methods.

Instantiations mainly focus on service potential: Not many instantiations have been found in the examined online research portal. Most of those existing instantiations found address the potential dimension. These instantiations mainly comprise lightweight online questionnaires and assessment tools, which can be implemented on tight budgets and do not require data integration with other application systems. They are often supposed to allow for a (semi-)automatic analysis of the participating company’s resources and service potentials. Other instantiations comprise modeling tools that are intended to support the conceptual development or engineering of new services and hybrid customer solutions. This service engineering approach is built on the premise that services can be engineered just like tangible products can. Service engineering traditionally is at the core of (manufacturing-focused) service research in Germany. Therefore, instantiations related to the potential perspective can build upon an extensive base of prior research. However, opportunities for the development of new instantiations also lie within the other perspectives. For instance, workflow applications could be used to make service processes run more efficiently and smoothly. Software support may also prove useful to let customers configure services by themselves so that the outcomes better meet their requirements.

The inside-out perspective dominates the outside-in perspective: Most of the IT artifacts support the potential perspective, whereas few artifacts are found to address the market and outcome perspectives. It can be argued that there is a dominating inside-out approach to the design of new services due to the manufacturing/industry focus of service science in Germany. This industry-stamped approach rather represents a technology-push than a market-pull mechanism. Therefore, a change towards a more customer-driven outside-in perspective still seems difficult to be accomplished for manufacturing companies as well as for researchers working in the field of SSME.

Constructs mainly focus on service outcomes: Innovative constructs are especially developed to describe the outcome perspective of customer solutions. This seems logical, since it is the integration of product and service components that is at the heart of research on hybrid value creation. Referring to the process and potential perspectives, existing constructs in terms of (process) modeling languages have already reached a considerable degree of maturity. These artifacts probably also have proven useful in service research and there has hardly been a need to develop additional service-specific artifacts for these perspectives.

Artifacts tend to be focused on isolated service phenomena and lack an integration of different perspectives: The cross reference matrix in Table 3 shows that few artifacts have been designed to holistically support several perspectives on services. Although it seems intuitive that narrowing the design problem to isolated – and perhaps more easily to grasp – phenomena speeds up the design process, this approach might also lead to a high degree of fragmentation of the results. This is disturbing, since companies require holistic solutions to support their everyday service business and cannot rely on a collection of isolated artifacts simultaneously. Therefore, one objective to be tackled in the future would be to integrate isolated artifacts with each other.

Artifacts tend to get stuck in a low degree of maturity and are seldom translated into applicable software instantiations: In addition to the apparent narrowness of IT artifacts in service science, their depth also seems quite limited, as can be inferred from Table 3. According to these figures, only a minority of artifacts is incorporated into software tools. This represents a barrier to the successful transfer of research results into practice, as constructs, models, and methods that only exist “on paper” might be difficult to apply in real-world business contexts.

5 Conclusions and Limitations

In this paper, we applied the framework for design science research in the service science discipline in order to analyze the IT artifacts that have been contributed to service science. We conducted an in-depth analysis of 123 research results taken from a German research portal that focuses on the research stream of hybrid value creation. We classified the 78 IT artifacts within this sample according to the service perspectives (potential, process, outcome, and market) and the artifact types (construct, model, method, and instantiation) distinguished by the framework. We discussed six key insights and gave some reasons for the observations made. Hot spots of design research within the service science discipline as well as opportunities for future research were disclosed.

Admittedly, the survey results and their discussion suffer from some limitations. The sample of artifacts analyzed can neither be labeled exhaustive nor representative. Our study relies on a research portal that is operated by a German research institution and which is open to all researchers and practitioners without further reviewing. The entries in this portal are dominated by German and governmentally funded research initiatives and thus do not represent the international service science community as a whole. Nevertheless, the portal represents the most exhaustive collection of research results – and especially of IT artifacts – available in the German-speaking service research community to date. Therefore, this study sheds light on the achievements generated in the recent 15 years of governmentally-funded service research initiatives in Germany and might also guide some directions for shaping future national as well as international research programs.

References

1. OECD: Enhancing the Performance of the Service Sector, http://www.value-chains.org/dyn/bds/docs/497/WolflOECDEnhancingPerformanceServicesSector.pdf#page=26
2. Spohrer, J.C., Riecken, D.: Service Science. ACM Commun. 49, 31–34 (2006)
3. Katzan, H.: Service Science: Concepts, Technology, Management. iUniverse, NY (2008)
4. Bundeszentrale für politische Bildung (bpb): Datenreport 2008 - Ein Sozialbericht für die Bundesrepublik Deutschland. Bonn, Germany (2008)
5. Spohrer, J., Maglio, P., Bailey, J., Gruhl, D.: Steps toward a science of service systems. Computer 40, 71–77 (2007)
6. Rai, A., Sambamurthy, V.: Editorial Notes - The Growth of Interest in Services Management. Inform. Syst. Res. 17, 327–331 (2006)
7. Bardhan, I.R., Demirkan, H., Kannan, P.K., Kauffman, R.J., Sougstad, R.: An Interdisciplinary Perspective on IT Services Management and Service Science. J. Manage Inform. Syst. 26, 13–64 (2010)
8. Gregor, S.: The Nature of Theory in Information Systems. MIS Quart. 30, 611–642 (2006)
9. Orlikowski, W.J., Iacono, C.S.: Research commentary: Desperately seeking the "IT" in IT research – A call to theorizing the IT artifact. Inform. Syst. Res. 12, 121–134 (2001)
10. Buhl, H.U., Weinhardt, C.: Die Aufgabe der Wirtschaftsinformatik in der Dienstleistungsforschung. Wirtschaftsinf 51, 469–542 (2009)
11. Velamuri, V.K., Neyer, A.-K., Möslein, K.M.: Hybrid Value Creation: A systematic review of an evolving research area. Journal für Betriebswirtschaft, forthcoming
12. Becker, J., Beverungen, D., Knackstedt, R., Matzner, M., Müller, O., Pöppelbuß, J.: A Framework for Design Research in the Service Science Discipline. In: Proceedings of the 15th Americas Conference on Information Systems (2009)
13. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quart. 28, 75–105 (2004)
14. Shostack, G.L.: How to design a service. European Journal of Marketing 16, 49–63 (1982)
15. Becker, J., Beverungen, D., Knackstedt, R.: The challenge of conceptual modeling for product–service systems: status-quo and perspectives for reference models and modeling languages. Inf. Syst. E-Bus. Manage. 8, 33–66 (2009)
16. Vargo, S.L., Lusch, R.F.: Evolving to a New Dominant Logic for Marketing. J. Marketing 68, 1–17 (2004)
17. Sampson, S.E., Froehle, C.M.: Foundations and implications of a proposed unified services theory. Prod. Oper. Manag. 15, 329–343 (2006)
18. Alter, S.: The work system method: connecting people, processes, and IT for business results. Work System Press, Larkspur (2006)
19. Alter, S.: Service system fundamentals: Work system, value chain, and life cycle. IBM Syst. J. 47, 71–85 (2008)
20. Edvardsson, B., Olsson, J.: Key Concepts for New Service Development. Serv. Ind. J. 16, 140–164 (1996)
21. Ramaswamy, R.: Design and management of service processes: keeping customers for life. Addison Wesley, New York (1996)
22. Jauch, L., Osborn, R., Martin, T.: Structured content analysis of cases: A complementary method for organizational research. Acad. Manage. Rev. 5, 517–525 (1980)
23. Elo, S., Kyngäs, H.: The qualitative content analysis process. J. Adv. Nurs. 62, 107–115 (2008)
24. Freelon, D.G.: ReCal: Intercoder Reliability Calculation as a Web Service. Int. J. of Internet Sci. 5, 20–33 (2010)
25. German Standards Institute (DIN): Publicly Available Specification 1094: Product-Service Systems – Value Creation by Integrating Goods and Services. Beuth, Berlin (2009)

Service Orienting the Swedish Vaccination Recommendation Activity with the Business Rules Centric Digital Service VacSam

Nicklas Holmberg, Odd Steen, and Sven Carlsson

Department of Informatics, Lund School of Economics and Management, Lund University, Ole Römers väg 6, SE-22363 Lund, Sweden
{Nicklas.Holmberg,Odd.Steen,Sven.Carlsson}@ics.lu.se

Abstract. Uniform control and coordination of immigrant children's vaccination is a critical current problem in the Swedish child health safety work. In this paper we discuss the Business Rules (BR) centric and SOA architected digital service VacSam. VacSam incorporates principles of SOA, Business Rules Approach, and Business Process Management. The incorporation is used for deriving VacSam from a part of the Swedish vaccination business process by separating decision logic from process logic. Based on regulatory texts and empirical investigations, VacSam BRs presently provides vaccination diagnosis of and recommendations to immigrant children. By ensuring the basic principles of SOA, VacSam becomes an eligible, SOA executable digital service. VacSam is in development and has hitherto been evaluated in an artificial context, where we show that the service can provide explained diagnosis of and recommendations to immigrant children's vaccinations totally based on natural language BRs.

Keywords: SOA, BRA, BP Modeling, Design Science Research.

1 Introduction

Design science research (DSR) should address three issues: 1) the nature of the artifact/problem/object studied, 2) research methodology, and 3) the research contribution [1].

The research presented here is part of a major DSR project addressing inter-organizational development of e-services. The specific service is a business rule-centric digital service for coordination of child vaccination (VacSam). Child vaccination is a global issue of increasing importance. Vaccinations are prescribed in accordance with schedules that vary per country. When immigrant children enter the Swedish health care system, it is problematic to coordinate vaccinations prescribed in previous countries and further vaccinations to be given according to the Swedish schedule. The VacSam project addresses this need by developing a rule-based digital service for vaccination schedule coordination by Swedish standards. Another contribution is that we show how a digital service can be developed based on SOA, Business Rules Approach, and Business Process Modeling. As part of the project, we also develop a user participatory method for inter-organizational development of e-services.

H. Jain, A.P. Sinha, and P. Vitharana (Eds.): DESRIST 2011, LNCS 6629, pp. 376–386, 2011.

Following Gregor and Hevner's [1] classification of DSR contribution types, this project's contributions are at level 1 and level 2. The VacSam systems is a level 1 contribution (artifact or situated implementation) and the two approaches, the SOA, BRA, BP modeling approach and the user participatory method for inter-organizational development of e-services are level 2 contributions (design principles—knowledge as operational principles/architecture).

Our research followed the guidelines presented in [2]: addressing a critical problem in a novel way, the use of justificatory knowledge, the process as a search process, evaluation of artifacts, etc.

The remainder of the paper is organized as follows. The next three sections present the justificatory knowledge and briefly present the SOA manifesto and basic principles. This is followed by a discussion on business processes and business rules. Section 5 presents the Business Rules Centric Digital Service (Vacsam). In Section 6, two evaluations are presented: 1) evaluation of the artifact instantiation (the VacSam system), and 2) evaluation of the understandability of the VacSam system's BRs. The paper ends with the conclusions in section 7.

2 Justificatory Knowledge

Service Science is where fundamental science and theories advocate innovation through Service Orientation. It indicates a movement from monolithic ISs towards loosely coupled responsible Digital Services (DS) [3, 4]. When a DS is derived from a part of a business, the basic principles of SOA are applicable and one or many Service Requestors could use the service-oriented resource. Increased service quality becomes a tentative result. A business could thus be designed, analyzed and formed as service-oriented and comply with the basic principles of SOA.

SOA can be realized with different technologies and functions independently from choice of realizing technology, e.g. Web Services, REST, and WCF [5, 6]. The criterion for SOA is therefore not about how one single product promises to redeem SOA, but how a specific product can be a part in reaching the desired level of service orientation in a business (see, e.g. [7]). Thus, business rules and business process modeling tools could correspond to SOA realizing products.

SOA brings decentralized ISs which components are loosely coupled, thus one or many digital services constitute an IS. As a result, increased variability is achieved as the components or digital services still knows about each other through common shared basic principles (see, e.g. [6]).

When realizing SOA, it is important to know which part of a business that is to be service oriented because its characteristics will affect the type of digital services designed. Consequently, it becomes a critical enabler to separate decision and process logic when designing, composing and categorizing a digital service (see, e.g. [8]).

Business Rules Management System (BRMS) remedies inconsistent business decisions providing the same properties for digital services transporting decision logic as digital services transporting process logic are equipped with in a SOA [9]. Based on that, separate digital services provide process and decision logic.

Advocating this new type of IS is one layer intended for decision logic (BRs) and one for application specific code or process logic [10].

The service requestor, provider and directory communicate through a transport medium constituting the conceptual basics of SOA. Thus, SOA is about service orientation of businesses and services on higher level of abstraction than SOA realizing technologies (see, e.g. [5]).

The SOA Manifesto [5] corresponds to service-oriented business design guidelines, i.e. high level SOA. The manifesto is based on six factors and constitutes 14 guidelines expressing: respect the social power structure in the business; be aware that the extent of SOA may vary and that products and standards cannot realize SOA alone; identify services by having business owners and engineers cooperate to verify that services are in parity with business goals.

The SOA Manifesto is realized through the basic principles of SOA intrinsically expressing: separation of concerns, encapsulation and information hiding.

2.1 Distinguishing Digital Services by Their Origin Business Process

Designing a digital service includes categorizing and preparing it for composition. Manual composition corresponds to workflow design addressing technical considerations. Automated composition corresponds to user defined expected goals (see, e.g. [11, 12]). Thus, digital service composition implies separating "what" from "how" (see, e.g. [11, 13]).

Declarative programming specifies "what" while imperative programming specifies "how" (see, e.g. [9, 11]). Thus, automated composition relates to declarative programming and manual composition to imperative programming [11].

As SOA addresses non-technological considerations [5, 6] it can identify functional requirements but not explain "how" these are realized imperatively (see, e.g. [5]).

Entity services hold a low transformative capability providing base information. Base information in turn is low transformative depending on the transformative nature of its origin BP.

As a result, entity services, activity services or process services could be categorized as bounded resources (existing before requested) or unbounded resources (created on request) depending on the transformative nature of their origin business process (see, e.g. [11]).

3 Business Rules

Business Rules (BR) represent decision logic as a rational and stateless set of interconnected rules leading to a decision based on known values of terms and facts producing logic values of true or false (see, e.g. [8, 14]). Business Rules should be based upon facts. In addition facts should be based upon concepts represented by terms (see, e.g. [15]). Morgan [14] defines a Business Rule accordingly:

> […] the conditions under which a process is carried out or the new conditions that will exist after a process has been completed [14, p. 59].

BRs express "what" should be done, or "what" is constraining a business activity from executing and not "how" it should be done (see, e.g. [15]).

BRs are integrated in the Enterprise Model (EM) since business rules approach strengthens the relations between business goal and vision [16, 17]. Thus, BRs are anchored in the EM by constituting the business rules model. The EM depicts how the business process model expresses "how" to achieve common goals, while the business rule model defines "what" to achieve common goals (see, e.g. [16]). The resource and actors model focus on actors performing "what" and "how". The concepts model establishes the common business vocabulary constituting business ontology [16].

Most BRs could be found throughout a business e.g., in legacy IS or in the business plan. Thus, business analysis is useful when discovering business rules [9, 13, 14]. That also implies to discover the business ontology and set the ground for the business vocabulary ensuring BRs to include the right terms and facts (see, e.g. [14]). Hence, the ontology becomes part of the business architecture.

3.1 Business Rules Approach

Business Rules Approach (BRA) provides abilities to declaratively express natural language, well formed, thus, atomic IS executable business rules (see, e.g. [9, 14, 18]). BRs should exist independent from workflows and procedures [15] spanning above business processes [18]. Based on that, BRA is an approach providing guidance in how to reset influence of decision logic from engineers to managerial business owners [9, 14]. BRA expresses BR design and management as an independent discipline of ISD [15, 19].

Business Rules Management System (BRMS), Business Rules Engine (BRE) and the BR repository are key components within BRA. The components look after BRs to express "what" and separate decision logic from application specific code (see, e.g. [9, 16]).

The design of digital services relates BRA to SOA (see, e.g. [9]), why service oriented concepts are shared between BRA and SOA [9]. Such concept is, e.g. that each BR set is supplied with a Web Service Description Language (WSDL) definition enabling the BR set to be requested as an independent Digital Services.

4 Business Process Modeling and Digital Service Categorization

Business process logic is a rational and stateful workflow of events, activities, actors, and decision points, transforming input to output. A Business Process (BP) is a collection of logically staged activities which together produces a value [8, 20]. Therefore, the distinction of a BPs characteristic, e.g., state and capacity, is what provides understanding of a BP [21].

The Business Rules Manifesto [18] advocates BRs to span and act governing over BPs separating decision and process logic. That provides loose coupling between BPs and BRs and between digital services and business processes. Hence, knowledge about the BP becomes fundamental when realizing SOA (see, e.g. [18, 22, 23]).

Business Process Modeling (BPM) provides abilities to understand "how" an event turns into an activity or "how" business activities execute and under which conditions [8]. Hence, that is to understand the decision logic governing the process logic in a business process.

BPM is thus applicable when isolating transformable segments from non-transformable in a BP, or transformable process logic from non-transformable process logic. The non-transformable segment constitutes, e.g., entity services or bounded resources providing the business with base information. Quite contrary, transformable segment constitutes, e.g., process services or unbounded resources providing the business with transformative information (see, e.g. [11]).

5 The Business Rules Centric Digital Service VacSam

The problem relevance of VacSam is to service-orient the Swedish vaccination recommendation activity by incorporating SOA, BRA and BP modeling. The design of VacSam corresponds to a technology-based solution solving a real world important business problem (see, e.g. [24]).

VacSam is derived from the process and decision logic of the Swedish vaccination recommendation activity [25]. Hence, VacSam acts as a DSS, which, based on BRs for vaccination schedules and known data of a child's vaccination history, will provide GPs with a diagnosis of vaccination status and recommendation for further vaccinations as compared to the Swedish schedule. VacSam also provide explanations of the outcome based on the rules that were fired. It will thus be possible for GPs to trace the conclusion back to the sequence of rules that gave rise to the result. Through this, GPs or vaccination experts can assess the credibility and applicability of the result.

VacSam is implemented using IBM Websphere ILog JRules suite of tools. JRules is a commercial full-fledged BRMS supporting rule-authoring, execution, and testing. The design process, described and discussed in [8, 25, 26], is made up of three stages which together lay the foundation for VacSam. In essence, we have worked with static analysis and rule analysis workshops, business process modeling and business concepts modeling (see, e.g. [9, 13, 14, 15]). Presently 1, 077 rules are implemented in the VacSam e-service.

5.1 Business Process Modeling

Using Event Process Chain (EPC) in ARIS Express 2.2 BP modeling was conducted in parallel with BR design. The BP was modeled based on empirical findings (see, [26]).

The Swedish vaccination business process has two overall states, basic and fully vaccinated, consequently not fully and not basic vaccinated are two more overall BP states. BP modeling provided noteworthy insights affecting the BR and DS design.

Understanding the BP by visualization made process logic transparent expressing "how" to calculate, e.g., the emigration age of a foreign child, which is vital for the recommendation provided.

Also, knowledge about the BP provided understanding for what the BRs govern; e.g. the conditions for executing the necessary calculation: `if` the person's age in months at the person's emigration date is at least 3 then vaccinate with DTwP. This calculation is an imperative JAVA expression in the execution class, but invoked by governing declarative business rules.

5.2 Business Rules Modeling

We performed a static analysis of regulatory texts from The National Board of Health and Welfare [27] and other sources on vaccination important to discover the business rules for VacSam, such as an Excel file with all national schedules provided by World Health Organization [28]. We transformed this into currently 1301 unique and atomic well formed BRs constituting the BR repository.

When we implemented the BRs in the JRules Business Action Language (BAL), we strived to resemble the structure and wording in the natural language BRs as much as possible. However, rules written in BAL follow the production rule structure of `if-then {-else}` that is quite reversed compared to the used rule structure from [14, 15] which are more linguistic, putting the subject and the truth value before the conditions.

5.3 Term-Fact Business Objects Modeling

In JRules, the business term and fact model is the foundation for BRs. Hence, all the BRs you author and run in JRules acts on the terms and facts. The term-fact model in JRules is the so-called Business Object Model (BOM), which holds the JRules equivalent of the executable classes. The members (attributes and methods) in the BOM hold the data and the functions that conditions and actions in the rules use. The BOM in the VacSam artifact is presently made up of these five classes: `CombinationVaccine`, `Person`, `Vaccination`, `VaccinationStatus`, and `Vaccine`. `Person` and `VaccinationStatus` hold many Boolean values, which are set by the rules at runtime. Inference in JRules works differently than in e.g. Prolog since a rule in JRules cannot use the result of another rule as input, which is a major weakness. Instead we use these Boolean attributes to store the result of fired rules in the working memory. The rules can thus use them in the condition part.

The Business Object Model (BOM) should be verbalized such that the objects and their members (attributes and methods) are given a more business friendly wording. As an example, there is the need to calculate the age of a child when he or she left the emigration country. Only then is it possible to know at what age (in days, weeks, months, or years) a child left a foreign vaccination schedule to enter the Swedish.

To do this, methods are implemented in the execution classes, which calculate the differences between the birth date and the emigration date. The methods e.g. `Person.ageInYearsInEmigrationCountry(Date the_date)` are implemented in the class `Person`. A more business friendly verbalization is `the age in years of {this} at {0}` with `{this}` as a placeholder for `'the person'` and `{0}` as a placeholder for `'the person's emigration date'`. It is therefore possible to state an age condition as in the following rule (words in `''` are variables):

```
if
 all of the following conditions are true :

      - 'the person'´s age in years in emigration country at
        'the person'´s emigration date is at least 14
    - 'the person'´s country is one of { "Norway" } ,
then

   set the name of 'the vaccine' to "BCG";
   add 'the vaccine' to 'the vaccination';
   set the dose of 'the vaccination' to 1;
   add 'the vaccination' to 'the person'´s vaccinations ;
   add "dose 1 of BCG vaccine at 14 years of age" to the arguments of
   'the reason';
```

5.4 Business Rules Flows

In VacSam, the rule sets execute using the RetePlus (an IBM proprietary version of the original Rete algorithm) option in JRules. This algorithm calculates which rules to fire, in what order, etc. using a working memory and an agenda. Since fired rules can update the working memory and thus change the agenda, rules can be fired again making inference and “intelligence” possible. Not having to mind the order of rules is according to the ideas of BRA.

Still, in some cases the order of rules and rule sets need to be controlled at design time. There are for instance several likely ways to be vaccinated against Diphtheria, Tetanus, and Pertussis (DTP). If the rules can establish that a child has been fully vaccinated against DTP according to the Swedish schedule, there is no need to also check if the child has been vaccinated against DT, since that is part of the already established DTP vaccination. Thus, we designed the rule flows in VacSam to avoid this situation.

5.5 Explanations of the Useful Final Result

In VacSam, there is a part in the rule’s action segment, which adds a text to something called `the arguments of 'the reason'`. One of the main goals of the VacSam service is that it can provide simple explanations of the outcome of rule execution. These explanations should not be the name of the fired rules, interesting as that might be, but part of the business rule itself. We therefore created a new class called Reason and added three array lists to this, which hold arguments, conclusions, and the top-level conclusion. We also provided an input/output parameter to be able to have the reasons as output from the rule engine.

The arguments hold the dose, vaccine and vaccination age set by all the rules fired for international schedules. These rules are executed first and the results produced are then compared to the rules for the Swedish schedule. In this case, the vaccination set by the rules can infer not vaccinated, partly vaccinated, or fully vaccinated with a vaccine or combination vaccine according to the Swedish schedule. The results of these rules are stored as conclusions, which are finally compared to the top-level decision. The outcome of this decision is stored in top-level conclusion. The concluding diagnosis and recommendation part of an explanation concerning a 13-year-old girl from Belize is shown below:

```
==== THE DIAGNOSIS IS ====
The child is considered as not fully vaccinated according to the Swedish
schedule since:
   a. ** The child is partly vaccinated or not vaccinated against HPV
   b. The child is fully vaccinated against Diphtheria, Tetanus,
      Pertussis, Hib and Polio
   c. The child is fully vaccinated against Measles, Mumps, and Rubella
   d. The child is a girl
   e. The child was born before year 2002

   ==== THE RECOMMENDATION IS ====
```

```
1 dose of Diphtheria vaccine should be given to the child since:
        a. The child is alien
        b. The child is at least 12 years old
1 dose of Tetanus vaccine should be given to the child since:
        a. The child is alien
        b. The child is at least 12 years old
The child is a risk child and may be given vaccine against Hepatitis B
```

6 Evaluation

Evaluation is a crucial component in an IS design science research process [2]. Evaluation can be done in a number of different ways using different methods and techniques [2, 29]. At this stage of the overall project we did two evaluations: 1. evaluation of the artifact instantiation (the VacSam system), and 2. evaluation of the understandability of the BRs.

An artifact instantiation can be evaluated in a natural or artificial context [30]. The former means, e.g., evaluation of VacSam in its intended context. At this stage of the project, it was deemed unrealistic to do naturalistic evaluations. Consequently, we did artificial evaluations. We did two different evaluations. The first evaluation focused on the VacSam system per se. This evaluation should evaluate if the system produces correct results. The evaluation method used was experiment. In the experiment different "scenarios" were generated. The scenarios were different children (the target group for VacSam) with different characteristics in terms of birth age, sex, birth country, emigration age, and immigration age. These are the critical characteristics that a user (e.g. a GP) can input to VacSam. We ran the different scenarios and evaluated if the system produced correct results. We started with simple scenarios and later increased the complexity. The overall result of the experiment was that the system produced correct results. As expected, during the experiment we got results that were correct per se, but not in conjunction with clinical praxis. Thus further analysis with VEs and GPs and empirical findings were needed. Results of this type led to redesign of VacSam. The VacSam evaluation was in part continuous and was a part of the design research method as briefly described in Section 5.

The second evaluation was related to the classical concern regarding user participation in requirement analysis and user participation in systems change. One of the goals was that when the VacSam systems will be in use it is expected that domain experts should be able to change the system. Future changes will be necessary as regulations etc. are changed and new vaccines are approved and recommended corresponding to the low transformative decision logic VacSam encapsulates. In this evaluation we also opted for an artificial evaluation. During the development process we found it necessary to test VacSam (i.e. the BRs) as we developed it. As we have discussed elsewhere [25] this was done in workshops with an immunization expert. The expert had no prior experience of business modeling for IS, including BR modeling, and was not experienced in ISD. Nevertheless, the expert had no problem understanding the rules and was able to approve and correct the rules, as well as amend and suggest new rules.

The evaluation suggests that it should be possible for a domain expert to understand the rules and also to make changes in the current rules as well as develop new rules without extensive training and support. Once VacSam is in use it should be interesting and critical to do naturalistic evaluations as well as evaluations of the other output of the overall project.

7 Conclusion

We have shown that VacSam as an artifact fulfills the ideas of the BRA largely, by a repository of business friendly BRs understandable to business people and separation of BR execution from other software in an IS system. Presently the BRs implemented in VacSam do indeed produce the required results of diagnosis and recommendation of vaccinations of immigrant children in Sweden. The results produced are entirely based on regulation texts and vaccination schedules transformed into BRs, and BR elicitation workshops with vaccination experts. The BRs are implemented and run in a SOA environment built on a commercial BRMS. The VacSam service produces results that should be understandable and traceable for health care workers and vaccination experts, and thus serve as the intended and much needed DSS for the vaccination activity in the Swedish health care sector.

Hence, VacSam is derived from process and decision logic. Also VacSam is an entity service, providing non-transformable base information and could therefore be categorized as a bounded resource existing before it is requested. In addition, VacSam is a stateless service and does not store requests on the server side. As a result, VacSam answers to separation of concerns, information hiding and encapsulation. Based on that, VacSam is an eligible digital service in line with the SOA manifesto and SOA basic principles and is thus allowed to execute in the digital ecosystem of SOA.

VacSam is also not a laboratory experiment, but a sincere effort to try and support a very important part of the work towards child health safety. Hence, VacSam needs to be implemented in the real world it is suppose to support. When it is and, hopefully, has been in use it will be interesting and critical to perform naturalistic evaluations as well as evaluations of the other output of the overall project.

References

1. Gregor, S., Hevner, A.: Introduction to the special issue on design science. Inf. Syst. & E-Bus. Mgmt. 9(1), 1–9 (2011)
2. Hevner, A., Chatterjee, S.: Design Science Research in Information Systems. Springer US, Heidelberg (2010)
3. Bitner, M.J., Brown, S.W., Goul, M., Urban, S.: Services Science Journey: Foundations, Progress, and Challenges. In: Hefley, B., Murphy, W. (eds.) Service Science, Management and Engineering Education for the 21st Century, vol. 3, pp. 227–233. Springer US, Heidelberg (2008)
4. Paulson, L.D.: Services Science: A New Field for Today's Economy. Comp. 39(8), 18–21 (2006)
5. Arsanjani, A., Booch, G., Boubez, T., Brown, P., Chappell, D., deVadoss, J., Erl, T., Josuttis, N., Krafzig, D., Little, M., Loesgen, B., Thomas Manes, A., McKendrick, J., Ross-Talbot, S., Tilkov, S., Utschig-Utschig, C., Wilhelmsen, H.: The SOA Manifesto, http://www.soa-manifesto.org/ (accessed September 30, 2010)

6. Erl, T.: SOA: principles of service design. Prentice Hall, Upper Saddle River (2008)
7. Erl, T.: SOA Principles, SOA Systems Inc., `http://www.soaprinciples.com/` (accessed March 4, 2010)
8. Holmberg, N., Steen, O.: Extracting Process Logic Needed In Business Rules Approach By Visualizing The Business Process Model. In: The 19th European Conference on Information Systems (submitted)
9. Graham, I.: Business rules management and service oriented architecture: a pattern language. John Wiley, Chichester (2006)
10. van Vliet, H.: Software engineering: principles and practice. Wiley, Chichester (2008)
11. Marinos, A., Krause, P.: What, not how: A generative approach to service composition. In: 3rd IEEE International Conference on Digital Ecosystems and Technologies, pp. 115–120 (2009)
12. Peristeras, V., Tarabanis, K., Goudos, S.K.: Model-driven eGovernment interoperability: A review of the state of the art. Comp. Std. & Interf. 31(4), 613–628 (2009)
13. Date, C.J.: What not how: the business rules approach to application development. Addison-Wesley, Reading (2000)
14. Morgan, T.: Business rules and information systems: aligning IT with business goals. Addison-Wesley, Boston (2002)
15. Ross, R.G.: Principles of the business rule approach. Addison-Wesley, Boston (2003)
16. Bajec, M., Krisper, M.: A methodology and tool support for managing business rules in organisations. Inf. Syst. 30(6), 423–443 (2005)
17. Rosca, D., Greenspan, S., Wild, C.: Enterprise Modeling and Decision-Support for Automating the Business Rules Lifecycle. Aut. SW. Eng. 9(4), 361–404 (2002)
18. BRG, The Business Rules Manifesto, Business Rules Group, `http://www.businessrulesgroup.org/brmanifesto.htm` (accessed April 23, 2010)
19. von Halle, B.: Business rules applied: building better systems using the business rules approach. Wiley Computer Publishing, New York (2002)
20. Zur Muehlen, M., Indulska, M.: Modeling languages for business processes and business rules: A representational analysis. Inf. Syst. 35(4), 379–390 (2010)
21. Smith, H., Fingar, P.: Business process management: the third wave. Meghan-Kiffer Press (2003)
22. Jun-Jang, J.: Service-Oriented Business Performance Management for Real-Time Enterprise. In: The 8th IEEE International Conference on and Enterprise Computing, E-Commerce, and E-Services, The 3rd IEEE International Conference on E-Commerce Technology (2006)
23. Larson, P.: Speak Softly and Carry a Big BPM Suite. KM World (2007) (accessed January 31, 2011)
24. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MISQ 28(1), 75–105 (2004)
25. Holmberg, N., Steen, O.: Better Support for User Participation using Business Rules Approach? In: 19th International Conference on Information Systems Development, ISD 2010 (2010)
26. Holmberg, N., Steen, O.: Business Rules Friendly or not so Business Rules Friendly Business Concepts Modelling - Early Experiences from a Business Rules Project on a Digital Vaccination Recommendation Service. In: The 33rd Information Systems Research Seminar in Scandinavia (2010)

27. Socialstyrelsen, Socialstyrelsens föreskrifter om ändring i föreskrifterna (SOSFS 2006:22) om vaccination av barn. Socialstyrelsen, Ändringsföreskrift, SOSFS 2008:7 (M) (2008), http://www.socialstyrelsen.se/sosfs/2006-22/Documents/2008_7.pdf
28. World Health Organization, Immunization surveillance, assessment and monitoring, http://www.who.int/immunization_monitoring/diseases/en/ (accessed March 17, 2010)
29. Carlsson, S.A., Henningsson, S., Hrastinski, S., Keller, C.: Socio-Technical IS Design Science Research: Developing Design Theory for IS Integration Managament. Inf. Syst. & E-Bus. Mgmt. 9(1), 109–131 (2011)
30. Baskerville, R.L., Pries-Heje, J., Venable, J.: Soft design science research: Extending the boundaries of evaluation in design science research. In: 2nd International Conference on Design Science Research in Information Systems and Technology (2007)

A Meta-model-Framework for Structuring the Requirement Analysis in Process Design

Stephanie Meerkamm and Stefan Jablonski

Chair of Applied Computer Science IV, University of Bayreuth, Universitätsstr. 30, 95447 Bayreuth, Germany
{stephanie.meerkamm,stefan.jablonski}@uni-bayreuth.de

Abstract. Today, with process management generally established as an integrated management tool, there is a strong interest in process modeling. The organizations spent time and effort in generating the optimal process model. Although there is a multitude of process modeling techniques available, the modeling process is often inefficient and the results are not satisfying. As each use case makes other demands on a process modeling language and tool it may be due to the neglect of some aspect of the design phase supposed to include a comprehensive requirement analysis and the implementation of these in an appropriate language and tool. Thus we want to offer a framework which focuses more on the design phase. The approach is based on a meta model hierarchy, focusing on modeling. This hierarchy is extended with a design phase. The increased quality of the final process models will also influence the whole process life cycle.

Keywords: process management, process design, process modeling, meta model hierarchy, specification of process modeling languages, requirement analysis.

1 Introduction

In order to produce a metal component with a certain level of granularity an appropriate tool is necessary. In this context "appropriate" means that the tool applied with a certain method, has to have the functionality to realize the fine tuning that results in the desired level of granularity. It has to be borne in mind that it is also the characteristics of the material that will influence the result of the tuning process; not all material can be machined to the same degree of granularity.

We learn from our example that although all requirements are derived from the characteristics of the final product, they can be related to different objects concerning the implementation: to the input, the applied tool and method and the output (i.e. the final product). In order to finally guarantee a high quality production of the output an initial comprehensive requirement analysis is essential. During implementation the specified requirements must not be neglected for example due to time and cost restrictions or other reasons.

In the context of process management, the overall aim of our research is to focus on the production of qualitatively good process models (i.e. output). Qualitatively

H. Jain, A.P. Sinha, and P. Vitharana (Eds.): DESRIST 2011, LNCS 6629, pp. 387–397, 2011.

good means, that all relevant characteristics of a process are modeled. Adequate modeling languages and tools are the crucial drivers for this. What is about the quality of a process model for a production process being unable to indicate the manufacturing and exposure time? For this, this so called design phase, which incorporates the requirements analysis and is done before modeling, is all the more important [1] [2]. Wrong decisions in the design phase (e.g. selection of an inadequate process modeling language) are difficult to compensate for in the following phases of the process life cycle, as modeling or monitoring, controlling (cf. [3], [4], [5]). With this the design must not be neglected.

In our opinion especially requirements for the applied tool, language and the final product must be identified and distinguished. We do not want to blame standard process modeling languages like BPMN [6] and UML [7], but it is most probable that some of the required characteristics of an individual process cannot be modeled with these. On the other hand, it is obvious that the usage of a domain specific process modeling language also has its drawbacks; among others, portability and exchangeability of process models will be reduced. Thus, a fine balance has to be drawn to decide which modeling language best fits the requirements [8] [3] [5].

In order to guide the process designer during the design phase and the requirement analysis we offer a framework. To this end, we base our approach on a meta model hierarchy defining a framework for the definition and usage of processes modeling languages (and tools). Extended with an explicit design phase the requirements can be structured according to the different levels.

In Section 2 the paper continues with an introduction into the process life cycle in order to give an idea about design and modeling and the context. Section 3 gives an overview of different design approaches. We introduce the meta model hierarchy as basis for our approach in Section 4. The way in which design and modeling can be integrated into this method is explained in Section 5, while Section 6 presents our new ideas in greater detail. Section 7 provides a brief evaluation. Section 8 concludes the paper.

2 Process Life Cycle

The design phase is an integral part of the process management approach. Process management constitutes a management concept to plan, guide and organize a company efficiently. It aims at the target-orientated management of time, quality and costs to achieve both strategic and operative goals. It can be illustrated as a so called "process life cycle" ([9], [10], [11], [12], [13], [14], [15]) which generally includes the following phases (see also Fig. 1):

Strategy: At first, the company's strategy for the achievement of its goals has to be defined. The strategy serves as a framework for all its business activities. It includes as well the definition of the business model and its organizational structure. It should be pointed out, that the strategy does not have to be re-defined after each process cycle. It serves more as a basic fundament for the whole process life cycle.

Design: The design phase comprises the collection of the requirements regarding the process models. Based on this the appropriate process modeling language is determined and an appropriate process modeling tool is selected. It might be that the process modeling language (and the tool) must be adapted to reflect specific requirements of the application domain. In addition, modeling rules must be specified dealing with the layout and structure of the processes.

Modeling: The process models are defined. This is done by applying the selected process modeling language and tool and the specified modeling rules. Produced models should be validated before being executed.

Execution: The process models have to be installed in the organization. The method of doing this will depend on the type of business. It may range from the publication of the process on the intranet to the deployment of the process in a fully automated workflow management system.

Monitoring and Controlling: For quality management reasons the processes must be monitored during execution. The target performance comparison aims at revealing deviation from the plan data. In doing so, corrective measures can be applied and experience gained from executed processes should be applied to improve subsequent process executions. The latter one results in a flow of continuous improvement as the process life cycle repeats starting again with the design (and in rare cases with the strategy).

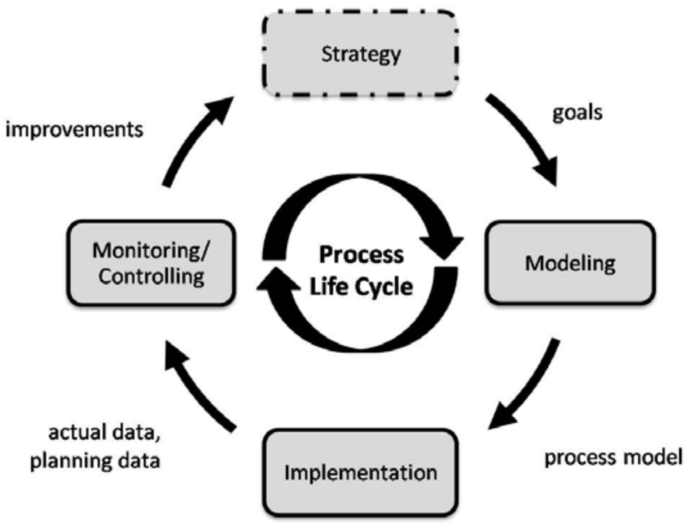


Fig. 1. Process Life Cycle

3 Related Work

Analysis of the literature shows that there is both a theoretical and a pragmatic approach to process design. In this Section we shall give a short overview of both.

Theoretical process design: The theoretical process design (see [16], [17], [18], [19], [20]), also called scientific design, is based on formal methods, models and techniques and the application of analytical procedures. As a design from scratch it aims to derive an optimal process layout, for example concerning the right order of the process steps. Analytical methods as *linear programming, branch-and-bound method* or *algorithmic procedures* are widely used in this context [16]. We not want to go into detail, because they are not in our focus. This also applies to the mathematical models as *Petri-nets* [20]. Another example is the so called *Product based design (BPD)* [18]. Starting with the analysis of the product, its structure of the building elements reveals which process steps are required to produce the object. Each process step of the resulting process model can be mapped to a part of the product. With this procedure, an effective and output-oriented process design can be realized.

The advantage of all these theoretical approaches is the structured and analytical procedure involved. As they are mostly proved and tested and developed over the time, they offer a well founded concept for the design. Furthermore, due to an explicit analysis the design decisions are traceable. Nevertheless, there are some disadvantages: Mostly the cost-benefit-ratio is disproportionate and they are difficult to apply with real world scenarios. The latter is due to the fact that it is difficult to formally characterize the elements and the context of the use case which have to be designed. Last but not least the process design is defined by lots of criteria which have to be balanced. The compromise which has to be made can only be developed individually but not with a formal method.

Pragmatic process design: The pragmatic process design (see [1], [21], [18]) is a practice oriented approach. The design is focused on existing processes which have to be improved. Weak points are eliminated, or at least minimized, which is mostly done by local updates. Thus it gets along with the process redesign as it is, for example, described in [22]. A well know and often applied method is *best practice* (see [1], [21]). A best practice is a historical and proved solution for a special situation. It is only applicable to particular circumstances and may have to be modified or adapted for similar circumstances. Meanwhile best practices are available in different industrial areas and divisions. The *method of benchmarking* (see [23], [24]) is based on the comparison of the own process with the processes of other, similar organizations. In case of deviation which leads to a disadvantage, the relevant processes have to be adapted accordingly.

A disadvantage of the pragmatic approach is that it does not necessarily lead to an optimal design. For this a solid analysis and synthesis of the processes is missing. Best practices are not generally applicable, they have to be adapted. However in general a pragmatic level of quality with a justifiable effort is achieved. Furthermore, these approaches are the fundament for the development of theoretical approaches of the process design and thus the basic for the overall design research area. They complement one another perfectly, thus a final judgment as to which approach is preferable should not be made.

4 Meta Model Hierarchy

As foundation for our ideas we use the meta model hierarchy according to [25], [26], [27] (we are aware that other approaches exist, see for example [28]). This meta model hierarchy is functionally comparable to MOF (Meta Object Facility) which is a generally accepted standard of OMG [29]. Within the meta model hierarchy process modeling languages, process models and process instances are defined at different hierarchical layers, namely M3, M2, M1 and M0 (see Fig. 2).

With this comes a structured and systematic procedure for the definition of process modeling languages and process models. The approach allows for the definition of (almost) arbitrary process modeling languages. This can be a standard but also a domain specific process modeling language.

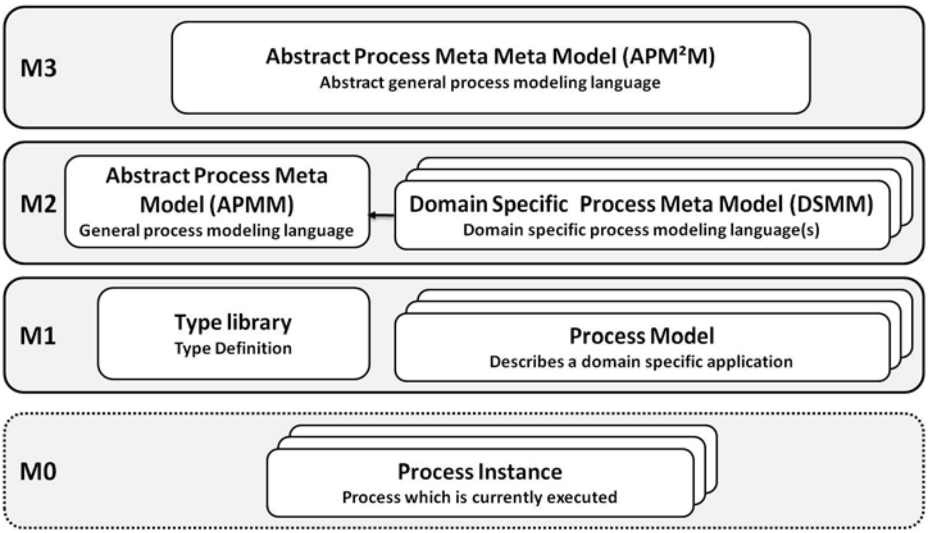


Fig. 2. Meta Model Hierarchy

In the following we shall give a short introduction into the meta model hierarchy. The focus is on modeling, thus the results of each level are recorded in a model. In order to facilitate understanding of the approach we do not explain it chronologically, but start on M1 with the definition of process models.

M1 – Process Model: On M1 the process models are generated. The process steps including further perspectives and characteristics are defined, i.e. data, organizations/roles and tools/ systems. All process elements are stored in libraries and can then be "re-used" to generate further process models. As an example, the process model "travel-reimbursement" is defined that describes the general procedure how an employee (organization/role) claims his / her travel expense (data).

M2 – Process Modeling Language: In order to define a process model on M1 a process modeling language must be available. This is provided on M2. At first a basic but still abstract process modeling language is defined which is represented by an abstract process meta model (APMM). It contains basic process modeling constructs such as process steps, control or data flow which are essential throughout all applications. In order to use the language in a special context a domain specific language can be derived represented by the domain specific process meta model (DSPMM). The domain specific customization is realized by adding new constructs, removing or adapting existing ones. For example, in the medical domain a special construct to depict medical decisions is added.

The most important fact to know in this context is that from the APMM user defined, domain specific process modeling languages can be derived.

M3 – Abstract Process Meta Meta Model: In order to define a process modeling language on M2 a basic language definition is required. This is called abstract meta meta model (APM²M). It comprehends the definition of basic modeling elements as for example nodes and arcs to form directed graphs.

M0 – Process Instances: The level M0 is not part of a modeling environment but of an execution environment since it contains the instances of a process which are executed. For example, the process model "travel-reimbursement" is instantiated to get the concrete process "travel reimbursement for Mr. Smith for his Asia-pacific trip in December 2009". It should be mentioned that the execution environment does not necessarily have to be an IT-related system; it can also be the organizational environment of the organization.

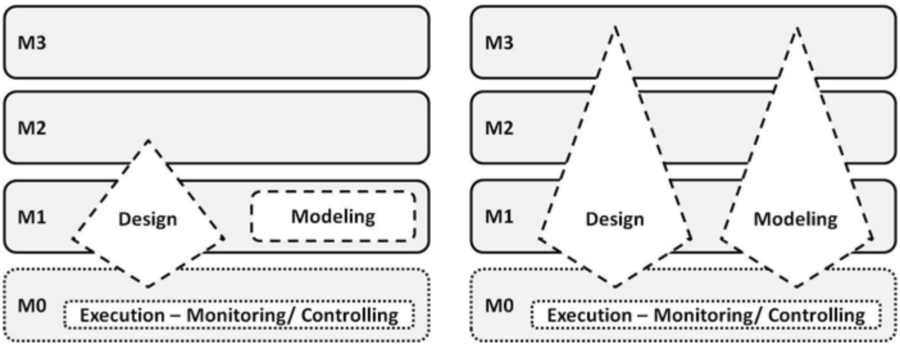
In the following (Section 5) we want to explain the way in which the phases design and modeling of the process lifecycle actually affect this meta model hierarchy. On the basis of this, we focus on the design and the ways in which the design phase can be extended and structured.

5 Design and Modeling in the Meta Model Hierarchy

The following figures show both the traditional way of integrating design and modeling in the meta model hierarchy, and the extended view (see Table 1). In order to develop a comprehensive approach execution and monitoring – controlling are also integrated into the framework on M0.

Traditional Interpretation: According to the definition of the process life cycle presented in Section 2, modeling is associated with layer M1 of the meta model hierarchy (see Table 1, A). Design is mostly seen as preparation of modeling by selecting a modeling language and defining modeling constraints. Thus it is mainly located on layer M1, too. Mostly the generation of domain specific model elements are neglected and standard process modeling languages are selected. This is indicated by referring the design phase to layer M2 only to a small extent. Design aspects referring to process execution on M0 are also considered. They are typically fewer than for modeling.

Table 1. Mapping Design and Modeling



(A) Traditional interpretation (B) Extended interpretation

Extended Interpretation: According to Section 4 modeling can be located at each layer of the meta model hierarchy. Each layer therefore produces specific models: M3: abstract process meta meta model - modeling primitives, M2: process meta model - process modeling language, M1: process model. On level M0 executable process models are generated. In consideration of this, it is possible to differentiate between different types of models.

The idea is now that each modeling (or model) level is extended with aspects of the design. Since these not only influence the definition of process models but also have an impact on the functionality and characteristics of a process modeling language, design is relevant to both layers M1 and M2; to the latter one to a much greater extend as it is actually the case. It might even be that very specific design requirements demand the definition of a completely new modeling paradigm on layer M3. With this interpretation the design is applied to all levels of the meta model hierarchy.

The result of the extended interpretation can be seen in Table 1, B, which illustrates the research contribution: The design phase should not just focus on process modeling on M1. Also, requirements relevant for the other layers of the meta model hierarchy must be considered. In view of our research goal to improve the quality of the process models, mainly M2 should be focused more. This allows an individual development of a process modeling language appropriate for a certain use case.

6 Categorization of the Requirements with the Meta Model Hierarchy

The idea of the hierarchical structure of design and modeling should now be transferred to the requirement analysis. Examination of the literature concerning the design of process modeling language shows that lots of requirements have already been identified (see for example [30] or [31]). It is not our goal to explicitly define additional requirements. This has to be done individually for each use case. In fact a framework for better structuring them should be offered (see Fig. 3).

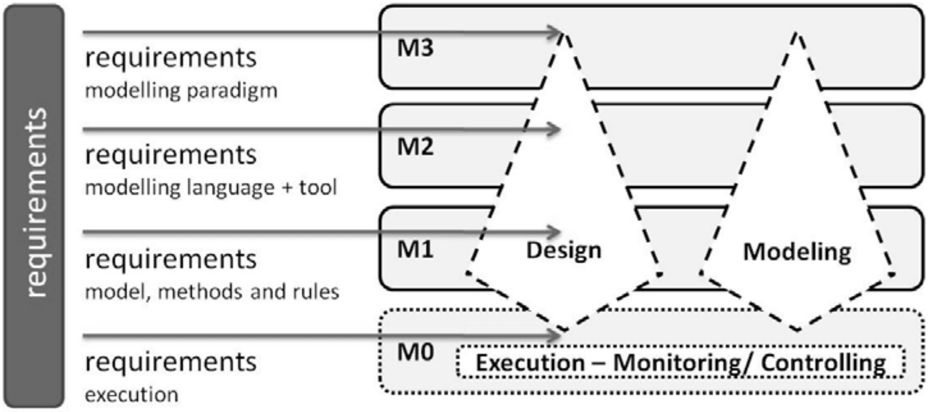


Fig. 3. Categorization of the requirements

In most cases requirements are initially collected without structure for example during a brainstorming session. Referring back to the extended interpretation of design and modeling (Table 1, B) the requirements can be categorized according to the different levels of the meta model hierarchy, which is illustrated in Fig. 3: M3: requirements regarding the modeling paradigm, M2: requirements regarding the modeling language (and the tool), M1: requirements regarding the process model and how the modeling has to be applied (method and rules), M0: requirements regarding the process execution.

With this, a precise definition of the requirements for a process model and the modeling language is possible as the basis for an adequate implementation. It is important to know that the different layers are not isolated from each other. The results from one layer are always be used on the next lower one.

7 Evaluation

For the purpose of evaluation we analyzed the requirements from two modeling projects at our chair; one from the medical, the other from the administrative domain. We identified several issues which could not be effectively modeled with modeling languages such as UML or BPMN and the associated tools. Examples are as follows:

(1) *Phases* which aggregate processes or tasks on a more abstract level than normal composite processes: The phases should be differentiated graphically from the latter - indicating also the time frame of the phases
(2) *Leaps* in the control flow which are not limited to one sheet of process: The relation between the starting and endpoint has to be illustrated and traceable from one sheet to the other
(3) *Variants*: The variation points, which indicated the difference between variants, have to be differentiated from normal gateways, connectors, etc.. In doing so they should not be limited to the control flow aspect, but also to others as, for example, the organizational or operational aspect.

Due to space limitation we are not able to enter into a detailed description of the individual problem statements, resulting requirements and elaborated concepts. However, for (3) a first prototype of an Open Meta Modeling Environment (oMME) (see [32] or [33]) could be generated. This is a platform for the development, adaption and visualization of meta models as it was described in Section 4. The platform should finally integrated an individually developed configuration concept which was published in [34].

8 Conclusion

In this paper we introduced an extended interpretation of design and modeling. The main goal is to improve the quality of the process models by means of optimal process modeling languages. This benefits modeling but also the subsequent phases of the process life cycle. During the design all requirements, viz. towards the modeling language, the tool, the model itself and also towards the execution of the final process, have to be collected. Doing this neither explicitly nor comprehensively, this could result in the selection of modeling languages and tools which do not really fit to the characteristics of the use cases. Due to the negative consequences along the whole process life cycle, we want to encourage the process modeler to conduct a comprehensive requirement analysis. With this he is able to select an individual process modeling language (and tool).

In order to support this idea we offered a framework for design and modeling. It is based on a meta model hierarchy used to structure design and modeling and the requirement analysis. With this, the process modeler should be able to specify an appropriate process modeling language to generate expressive and meaningful process models.

References

1. Hevner, A.R., March, S.T., Prak, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28, 75–105 (2004)
2. Glass, R.: On Design. IEEE Software 16, 103–104 (1999)
3. Recker, J., Indulska, M., Rosemann, M., Green, P.: Business Process Modeling - A Comparative Analysis. Journal of Association for Information Systems 10, 333–363 (2009)
4. Recker, J., Mendling, J.: Adequacy in Process Modeling: A Review of Measures and a Proposed Research Agenda. In: 8th Workshop on Business Process Modeling, Development and Support, Trontheim (2007)
5. Rosemann, M.: Potential Pitfalls of Process Modeling: part A. Business Process Management Journal 12, 249–254 (2006)
6. OMG: BPMN Business Process Modeling Notation (2010)
7. OMG: Unified Modeling Language (UML) (2010)
8. Moody, D.L.: Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. Data & Knowledge Engineering 55, 243–276 (2005)

9. Abecker, A., Hinkelmann, K., Maus, H., Müller, H.J.: Geschäftsprozessorientiertes Wissensmanagement – Effektive Wissensnutzung bei der Planung um Umsetzung von Geschäftsprozessen, vol. 1. Aufage Springer Verlag, Berlin (2002)
10. Allweyer, T.: Geschäftsprozessmanagement – Strategie, Entwurf, Implementierung, Controlling. W3L-Verlag, Herdecke (2005)
11. Becker, J., Kugler, M., Rosemann, M.: Process management. Springer, Heidelberg (2003)
12. Harrington, H.J.: Total Improvement Management - The Next Generation in Performance Improvement. McGraw-Hill, New York (1995)
13. Lee, R.G., Dale, B.G.: Business Process Management: A Review and Evaluation. Business Process Management Journal 4, 214–225 (1998)
14. Schmelzer, H.J., Sesselmann, W.: Geschäftsprozessmanagement in der Praxis, vol. 6. vollständig überarbeitete und erweiterte Auflage Carl Hanser Verlag, München (2008)
15. Weske, M.: Business Process Management - Concepts, Languages, Architectures. Springer, Heidelberg (2007)
16. Hofacker, I., Vetschera, R.: Algorithmical Approaches to Business Process Design. Computer & Operational Research 28, 1253–1275 (2001)
17. Reijers, H.: Design and Control of Workflow Processes: Business Process Management for the Service Industry. Springer, Heidelberg (2003)
18. Reijers, H.A., Netjes, M., Vanderfeesten, I.: On pragmatic and formal process design aproaches. In: 8th Workshop on Business Process Modeling, Development, and Support, Trontheim (2007)
19. van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M.: Business process management: A survey. In: van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M. (eds.) BPM 2003. LNCS, vol. 2678, pp. 1–12. Springer, Heidelberg (2003)
20. van der Aalst, W.M.P.: The application of Petri Net to Workflow Management The Journal of Curciuts. Systems and Computers 8, 21–66 (1998)
21. Reijers, H.A., Liman, M.S.: Best practice in business process redesign: an overview and qualitative evaluation of succesful redesign heuristics Omega (2004)
22. Davenport, T.H., Short, J.E.: The New Industrial Engineering: Informationtechnology and Business Process Redesign. IEEE Management Review (1990)
23. Malone, T.W., Crowstone, K., Lee, J., Pentland, B., Dellarocas, C., Wyner, G., Quimby, J., Osborn, C., Bernstein, A., Herman, G., Klein, M., O'Donnell, E.: Tools for Inventing Organizations: Towards a Handbook for Organizational Processes. Management Science 45 (1999)
24. Zairi, M., Sinclair, D.: Business process re-engineering and processmanagement - A survey of current practice and future trends in integrated management. Business Process Re-engineering & Management Journal 1, 8–30 (1995)
25. Aktinson, C., Kühne, T.: Concepts for comparing modeling tool architectures. In: Briand, L.C., Williams, C. (eds.) MoDELS 2005. LNCS, vol. 3713, pp. 398–413. Springer, Heidelberg (2005)
26. Aktinson, C., Kühne, T.: The essence of multilevel metamodeling. In: Gogolla, M., Kobryn, C. (eds.) UML 2001. LNCS, vol. 2185, p. 19. Springer, Heidelberg (2001)
27. Henderson-Seller, B., Gonzalez-Perez, C.: A power-type-based metamodeling framework. Software and System Modeling 5, 72–90 (2006)
28. Karagiannis, D., Kühn, H.: Metamodelling Platforms. In: Bauknecht, K., Tjoa, A.M., Quirchmayr, G. (eds.) EC-Web 2002. LNCS, vol. 2455, p. 182. Springer, Heidelberg (2002)
29. OMG: MOF 2.0 Specification (2009)

30. Frank, U.: Evaluating Modelling Languages: Relevant Issues, Epistemological Challenges and a Preliminary Research Framework. Arbeitsbericht des Instituts für Wirtschaftsinformatik, Koblenz, vol. 15 (1998)
31. Frank, U., van Laak, B.D.: Anforderungen an Sprachen zur Modellierung von Geschäftsprozessen. In: Frank, U., Hampe, F. (eds.) Arbeitsbericht des Instituts für Wirtschaftsinformatik, vol. 34. Institut für Wirtschaftsinformatik, Universität Koblenz-Landau, Koblenz-Landau (2003)
32. Volz, B., Jablonski, S.: Towards an Open Meta Modeling Environment. In: 10th Workshop on Domain-Specific Modeling (DSM 2010) Reno/Tahoe, Nevada, USA (2010)
33. Volz, B., Jablonski, S.: OMME - A Flexible Modeling Environment. Workshop on Flexible Modeling Tools (FlexiTools@SPLASH 2010), Reno/Tahoe, Nevada, USA (2010)
34. Meerkamm, S.: Configuration of Multi-Perspectives Variants. In: 1st. Int'l. Workshop on Reuse in Business Process Management (rBPM 2010), Hoboken, NJ, USA (2010)

Why Grandma Trims the Brisket: Resource Flows as a Source of Insight for IT-Enabled Business Process Design

George M. Wyner

Boston University School of Management
595 Commonwealth Avenue, Room 613
Boston, MA USA 02215
gwyner@bu.edu

Abstract. Systems analysts and organizational designers are increasingly called upon to rethink business processes both to respond to changing conditions and to realize the potential of new information technologies. Existing process modeling tools typically represent one particular version of a process but do not represent the alternative ways in which that process could be organized. Dependency diagrams offer analysts a way past this difficulty by representing the underlying coordination issues in a process, allowing analysts to consider alternative process designs. Unfortunately, dependency diagrams can be difficult to draw because dependencies can be difficult to discover. This paper describes Resource Flow Graph Analysis (RFGA), a method for developing dependency diagrams which leverages the observability of activities and resource flows to allow analysts to systematically uncover the dependencies which shape a given business process. The potential application of the method to process analysis and system design is illustrated by a "design exercise."

Keywords: process design, dependencies, coordination, resource flows, design methodology.

1 Introduction

A story is told of a young girl who is watching her mother prepare brisket (a kind of meat dish) in the kitchen. The mother carefully trims the end off the piece of meat before placing it in the pot to cook. The girl asks her mother why she has trimmed the end of the brisket and the mother replies, "That is how your grandmother taught me to prepare brisket."

Later the girl is visiting her grandmother's house and asks, "Grandma, why do you cut off the end of the brisket? Is that part not good to eat?"

Her Grandmother replies, "When your mother was young, we had a small kitchen and a very small pot for cooking brisket. I cut off a piece so it would fit in the pot."

While this story is somewhat whimsical, we actually see this phenomenon all too often in organizations: for example a business in which no one knows why the pink copy of some form gets filed or who uses a given report. We develop processes but

H. Jain, A.P. Sinha, and P. Vitharana (Eds.): DESRIST 2011, LNCS 6629, pp. 398–411, 2011.

lose touch with the forces that shaped them. Then, when the situation changes, we are not always clear about how a process can or should be changed.

The “brisket problem” is of great relevance to systems analysts in that business process analysis is closely integrated with information systems design [1, 2]. Process modeling helps to define the services a system ought to provide in order to add value to an organization and hence is a key part of requirements definition. In addition, by analyzing a business process, analysts can also discover new ways to organize work, reinventing business processes to take advantage of the capabilities of information technology [3-6].

It follows that in order to produce the most useful analysis and design, analysts must effectively resolve or avoid the “brisket problem.” Unfortunately, the process modeling tools we use to represent and analyze processes tend to be vulnerable to this pitfall:

- They do not show us what parts of the process can change when conditions change (e.g. when Grandma gets a larger kitchen) and what parts of the process must remain fixed because of fundamental constraints.
- They limit the set of possibilities we can consider and thereby leave the designer vulnerable to suboptimal design -- selecting the best from among a limited set of alternatives while never considering even better possibilities.

In this paper I will argue that a different approach to process representation is needed when engaged in rethinking a process: the “dependency view.” In the dependency view we make a distinction between the core *production activities* of a process and the *coordination activities* [6, 7]. P*roduction activities* represent the parts of a process which must remain the same. However things change in Grandma’s kitchen, the goal remains to prepare a brisket, which means that meat will need to be *obtained*, *prepared*, and *cooked*; these are the production activities of this process. *Trim end off meat* is not a production activity, as Grandma points out to her granddaughter.

Any left over (non-production) activities in any given version of a process may be *coordination activities*. Coordination activities are necessary in order to ensure that the production activities are combined in a way that yields good results. In Grandma’s original small kitchen, she needed to add the coordination activity *trim end off meat* in order that the cook brisket production activity could complete successfully. Such coordination activities may be said to manage the *dependencies* among the production activities of a process. I will give a more precise definition of dependency below, but for now we can think of a dependency as some set of constraints on the interaction between one or more production activities required for the goals of the process to be met. In the case of the brisket, there is a dependency between *obtain meat* and *cook meat*: the meat obtained by the former activity must fit in the pot used by the latter activity. We will refer to the set of coordination activities used to manage a given dependency as a *coordination mechanism*.

By viewing a process in this way, we can consider alternative methods for coordinating a process. For example, once Grandma has a larger kitchen she can consider purchasing a larger cooking pot, at which point the dependency between *obtain meat* and *cook meat* is satisfied without trimming the brisket and therefore that step in the process can be eliminated.

Malone et al [6] developed the dependency diagram (see Figure 4 for an example) as a process representation that shows production activities together with the dependencies among them. By showing the dependencies of a process rather than specific coordination activities, we can consider alternative coordination mechanisms for each dependency and thus consider how a process can be modified while retaining its core production activities. Dependencies define the requirements that must be satisfied for a process to continue to function. The intriguing possibility is that by representing a process using a dependency diagram we will be able to avoid the "brisket problem" or cope better with it.

One goal of this paper is to restate the case, made previously by Malone et al (absent the brisket), that the dependency diagram is a useful tool for process design and innovation. However, the primary contribution of this paper is to propose a solution to a problem that arises when one seeks to put the dependency diagram approach into practice:

Dependency diagrams can be difficult to draw because dependencies can be difficult to find. Dependencies are intangible aspects of a process: they are relationships among activities and thus cannot be directly observed. In particular, dependencies are typically only visible when they are not managed effectively. When a process is running smoothly it is not always evident what factors are contributing to this success. Thus to discover a dependency one must engage in counter-factual thinking: imagining what can go wrong in order to arrive at the constraints needed to avoid such problems.

A second factor that makes dependencies difficult to discover is the distributed nature of dependencies. A dependency exists between two or more activities. In order to understand the dependency one must understand the interaction among these activities. However, sometimes activities are carried out by different actors and in different locations and there may be no one person with a clear understanding of both sides of such an interaction.

Crowston and Osborn [8] describe two basic strategies for identifying dependencies: In *activity-focused analysis* the analyst looks for coordination activities and seeks to match them with the dependency they manage. This can be effective, but may miss dependencies that are managed informally or implicitly. For example, a weekly staff meeting may be used to manage multiple dependencies and it may be difficult to match that one activity with all those dependencies. This form of analysis would also miss dependencies that are currently unmanaged or managed by exception handling that does not show up in a process map.

Thus Crowston and Osborn also identify a second strategy: *dependency-focused analysis*. In this form of analysis the analyst searches for dependencies directly by examining the flow of resources among activities. This technique has the advantage that resources are often tangible and thus it is easier to identify resource flows. This idea, that resources can be used to discover dependencies, is the key insight on which the current paper builds. As will be explained in detail below, each dependency is associated with a specific pattern of resource flows and thus by examining the resource flows in a process we can uncover the dependencies.

In this paper I propose a method, *Resource Flow Graph Analysis* (RFGA), which can be used to support a systematic analysis of resource flows in order to discover dependencies. RFGA develops a map of the resource flows within a process and uses this map to discover dependencies.

RFGA differs from and adds value to the Crowston and Osborn approach in two respects: First, the approach introduces a representation, the *resource flow graph*, which is specifically designed to display and analyze resource flows. Second, where the Crowston and Osborn method first develops a list of activities and resources and then analyzes the flows among them, RFGA builds up a resource flow graph in a more iterative, emergent manner, allowing for the possibility that resources and even activities will be discovered during the process of constructing a resource flow graph.

The contribution of this paper will be to provide a description of RFGA and an example of how it can be used to construct a dependency diagram. The logic of the paper is that RFGA is a technique that analysts can use to discover dependencies and draw dependency diagrams and that analysts can in turn use those dependency diagrams to distinguish between the parts of a process that can vary and the parts that must remain constant. Analysts can thus systematically explore potential modifications to the design of a process and thus overcome the potential pitfall of the brisket problem.

This paper adopts the design science research paradigm [9] in that its contribution takes the form of novel artifacts and a preliminary assessment of the potential utility of those artifacts. To use the terminology put forward by March and Smith [10] and adopted by Hevner et al [9], this paper contributes two new artifacts: a new *model*, the resource flow graph, and a new *method*, RFGA. The utility claimed for these artifacts is that they facilitate the development of dependency diagrams, an existing design artifact whose utility in turn lies in supporting analysts in their efforts to design and improve business processes.

Section two of the paper reviews the theoretical basis for the dependency approach and describes its key representation: the dependency diagram. Section three introduces RFGA, explains its principle benefits, and describes the method in some detail. Section four illustrates the application of the method to a simple example. Section five discusses evidence for the usability and usefulness of the method. Finally, section six summarizes the contribution of RFGA to process modeling research and practice and describes directions for future research.

2 The Dependency View of Process

Before we proceed it will be helpful to provide a working definition for the term “process.” A *process* is a recurring pattern of behavior which is associated with some system and to which a purpose is ascribed. The purposeful actions which comprise a process are referred to as *activities* or *subactivities* (this latter to emphasize that individual activities are part of something bigger).

Systems design methods offer a number of techniques for modeling processes: flow charts and data flow diagrams are both used in the structured modeling approach [11] and, more recently, the activity diagram (an extension to the flow chart) is a part of the UML modeling specification [12, 13]. While these modeling capabilities capture important aspects of a process, all these approaches share the same important limitation: they represent one particular version of a process but do not represent the alternative ways in which that process could be organized. Activity diagrams do not distinguish between control flows that are required by business constraints and those that are arbitrary or even counter-productive. Data flow diagrams represent information flows but not whether they are essential to achieving the process goal or whether they persist for historical reasons.

Once a process has been designed, traditional process maps are very useful, but for the reasons just given, they do not provide much support for generating and evaluating alternative designs. These limitations suggest the need to represent the underlying business and technical constraints which must be taken into account in organizing activities into a successful business process. As noted in the introduction, Malone et al have proposed addressing the limitations of existing process models by representing the production activities of a process together with its dependencies [6, 7]. The basis for this approach is coordination theory.

Coordination theory begins with a definition: coordination is the act of managing dependencies between activities [7][1]. Associated with each dependency is a *coordination mechanism*, which consists of those components of the system which manage that dependency. For example, a shared resource dependency might be managed by a "first come first serve" policy or a market-like bidding mechanism or managerial fiat, and so forth [7, 16].

Three types of dependencies are identified in coordination theory: A *flow dependency* captures the issues which arise when a resource flows from the activity which produces it to the activity which consumes it. A *sharing dependency* identifies the issues which arise when a single resource is consumed by multiple activities. A *fit dependency* identifies the issues which arise when a single resource is produced through the joint action of multiple activities.

A dependency diagram (such as that shown in Figure 4 below) represents the dependencies associated with a process. The diagram includes activities, represented by rectangles, and their associated dependencies, represented by ovals. The direction of the links indicates whether an activity is producing (arrow points to dependency) or consuming (arrow points to activity) the resource associated with a dependency.

The dependency diagram omits any coordination activities since these will be associated with the management of one or more dependencies. By abstracting away coordination activities and information flows, a dependency diagram focuses attention on the dependencies which define what sort of coordination is required and what sort of information is needed to support that coordination.

Thus the dependency diagram addresses the limitation we have identified in existing process modeling techniques by: (1) distinguishing between the parts of the process that must be present in any redesign and the parts that may be modified, and (2) identifying the requirements that must be addressed by the variable parts of the process, that is, the dependencies these variable parts must manage.

Once a dependency diagram has been developed the analyst can proceed to consider potential coordination mechanisms for managing each dependency, taking into account new possibilities that may be enabled by new technologies or changing business conditions.

[1] This brief description of coordination theory is adapted from [14] and [15].

3 Resource Flow Graph Analysis

Hopefully at this point the reader is convinced that dependency diagrams have the potential to play a useful role in process design, but the value of this approach will depend on the ability to identify dependencies for inclusion in the diagram. As argued in the introduction, dependencies themselves can be hard to discover, both because they are not entirely observable and because they may require knowledge of a process that is distributed across multiple actors. Crowston and Osborn [8] have developed methods for discovering dependencies, a key component of which is the tracing of resource flows within the process. *Resource Flow Graph Analysis* (RFGA) is a method which has been developed to assist in this "resource-based" discovery of dependencies.

The theoretical basis for RFGA is the relationship between resources and dependencies originally articulated by Zlotkin [16]. For our purposes a resource is a physical or information object viewed as either an input or output of an activity. Resources flow from the activities which produce them to the activities which consume them. The three types of dependencies defined in coordination theory map naturally to three distinct patterns of resource flows as indicated in Figure 1.

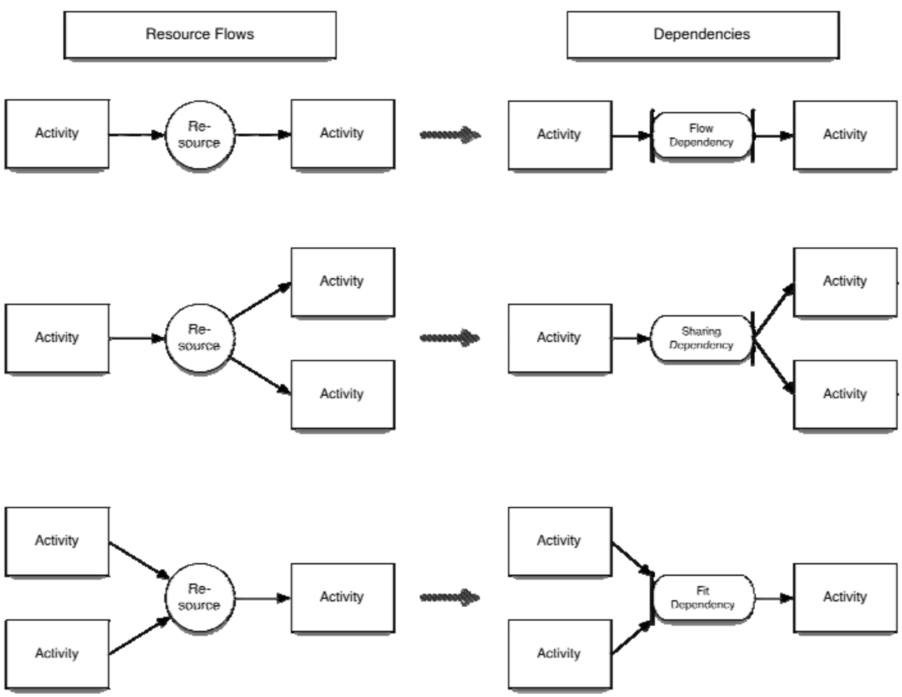


Fig. 1. Mapping between resource flows and dependencies

RFGA allows an analyst to begin by identifying activities and resources, which are generally more visible to process observers and participants, and then identify the relationships between these elements in order to construct a *resource flow graph*, a map of the resource flows within a process[2].

The resource flow graph includes both activities and the resource flows among them. In general an activity may both consume and produce resources and these relationships are represented as resource flows into and out of activities. An example of a resource flow graph can be seen in Figure 3 below. Note that this diagram corresponds to the dependency diagram depicted in Figure 4.

Once a resource flow graph has been developed, dependencies can be identified using the mapping given in Figure 1 above. Finally, the analyst can use process analysis to identify desired system functionality. Thus RFGA can be described as a sequence of two key objectives: (1) Use informal process descriptions to develop a resource flow graph. (2) Use the resource flow graph to discover dependencies and develop a dependency diagram.

3.1 Creating the Resource Flow Graph

The first task for the analyst is to construct a resource flow graph from informal process descriptions. This is made more difficult by the distributed nature of process knowledge. A process stakeholder may possess local knowledge -- knowledge of some set of activities and resources -- with only limited awareness of how these elements interact with distant parts of the process. This distribution of process knowledge across multiple stakeholders brings with it additional issues that may create difficulties in constructing a useful resource flow graph: (1) Process descriptions may reflect multiple and possibly conflicting points of view, especially concerning the goals of the process. (2) Even with multiple informants with overlapping knowledge, the resulting process descriptions may omit activities and resource flows.

RFGA addresses these issues by focusing the initial analysis on the identification of individual activities and resources without attempting to specify their interconnections fully. The working assumption is that the preliminary list of activities and resources may be incomplete and may include multiple, possibly conflicting, views of the process. These process components are then linked together by identifying which resources are consumed, accessed, produced, or modified by each activity. This integration of process fragments into a coherent resource flow graph supports the articulation of a point of view for this analysis. In addition this integration may focus attention on previously overlooked resources and activities. More specifically this phase of RFGA consists of the following steps:

1. Read through the process descriptions (i.e. existing documentation and analyst notes) and search for activities and resources.

---

[2] Portions of this section have been adapted from [14] and [15]. These earlier publications referred to the method as *Text Based Process Analysis* (TBPA). The phrase *Resource Flow Graph Analysis* (RFGA) has been adopted as better identifying the distinctive approach of the method. For a more extensive treatment of this method see also [17].

2. Add these components to a process fragment diagram (e.g. Figure 2 below). At this stage the activities and resources are for the most part not connected to each other and may be placed in an arbitrary order (often, as a matter of convenience, the placement typically reflects the order in which components were encountered).
3. Using this process fragment diagram, look for resource flows which may connect different activities together. As these flows are added to the process fragment diagram, larger fragments of the process begin to coalesce.
4. As the process fragments grow in size, examine them for an overall point of view. This often turns out to be a critical step in moving from the collection of process fragments in step 3 above to the resource flow graph that follows. Experience with RFGA suggests that it is often an explicit point of view that helps to select a subset of the process fragments and fit them together into a whole. Clearly there may be more than one process of interest and the same process may be viewed from several different viewpoints in the process fragment diagram. That is to be expected. One may in fact construct several process maps from a single text, or one may choose one particular point of view for one particular process and favor it over the others. What is important is that these choices and these issues be recorded for future review.

Note that there is an inductive bottom-up approach in which one connects fragments to build a resource flow graph, and at the same time there is also a top-down approach in which one develops a point of view which serves as an organizing framework for the fragments.

3.2 From Resource Flow Graph to Dependency Diagram

In the second phase of RFGA the analyst uses the resource flow graph to construct a dependency diagram:

1. The analyst removes from the resource flow graph any resource issues that are outside the scope of the current analysis.
2. The analyst also removes from the resource flow graph any activities and resources which are part of coordination mechanisms, as they will be represented by dependencies.
3. With the remaining resource flows, the analyst identifies dependencies using the correspondences shown in Figure 1. Note that, unlike a resource flow graph, which typically represents all the flows involving resources of importance to the activities, the dependency diagram represents instead only those dependencies which are of importance to the process; these critical dependencies may correspond to all or only some of the flows in the corresponding resource flow graph. A dependency might be considered unimportant by the analyst because its effect on the process outcomes of interest is insignificant or because it has a significant impact but is easily managed.
4. The analyst may choose to further simplify the dependency diagram by aggregating activities or by restricting the scope of the analysis. The purpose of this simplification is to make the dependency diagram readily understandable.

4 A Simple Example

To get a better sense of how RFGA is employed in practice, we will walk through a simple example based on one of several "design exercises" which were carried out to explore the usability and practical implications of this method. The focus of these exercises was primarily on producing a set of dependency diagrams, with a brief discussion of process insights and potential supporting systems. In these design exercises the process description we began with was provided in conversation with a participant, who brought an organizational issue involving process design, product design, organizational learning, or organizational change. At the close of each exercise I asked participants to provide feedback about the method.

This particular example was a design exercise carried out with Greg[3], who is a research scientist at Commonwealth Technical University (CTU), a research university in the eastern United States. Greg is part of the Innovation Group, which together with two other research groups at the University is being sponsored by CompCo, a leading computer component manufacturer. CompCo has asked the three research groups to jointly produce a prototype of a system to support business problem solving. The prototype will be tested on a supply chain management problem provided by CompCo. The three groups are to contribute to the prototype as follows: (1) The Hardware Group will provide the hardware platform for the prototype. (2) The Modeling Group will provide a model of the business problem. (3) The Innovation Group will provide a process analysis of the business problem. Our goal in this design exercise was to develop a description of the collaborative design process for the prototype.

Greg and I began by identifying the activities and resources associated with the design process, resulting in the process fragment diagram shown in Figure 2. We then identified resource flows and constructed the resource flow graph shown in Figure 3. In doing so, we eliminated several resources (*funds*, *software & hardware interfaces*, *research agendas*, *researchers*, *computing power*, and *other projects*) as being (tentatively) outside our scope. We identified *Construct Example* as a subactivity of *Acquire Resources*, and *Design Interfaces* as a part of *Integrate Design*, thus omitting explicit mention of these activities from the resource flow graph. We considered *Coordinate Research Effort* to be part of the coordination mechanism for this process and thus it, too, is not included here.

Finally, we added the activity *Integrate Design* and the resource *research prototype* because they are needed to complete the account of the process. The prototype, which we had omitted in Figure 2, is a key deliverable in this process and hence a critical resource. *Integrate Design* is the activity which produces that resource.

Based on this resource flow graph, we then developed a dependency diagram (Figure 4). We removed the leftmost activity shown in the resource flow graph, *Acquire Resources*, because we elected to consider any activity prior to the design task as outside the scope of our analysis. Accordingly, we included instead a flow dependency and sharing dependency for resources. We adopted the generic term "resources" to bring back into our analysis such resources as *computing power* and *funds*. We decomposed *Communicate Results* into the four principal communication activities

[3] In the account which follows, names and other identifying characteristics have been disguised or omitted to protect the confidentiality of the information disclosed.

associated with the sponsor and the three research groups. This allowed us to represent an additional fit dependency which Greg described as involving these four distinct communication activities.

This dependency diagram then served as a basis for a discussion about the choice of coordination mechanisms for this process. For example, the fit dependency between the three design activities and *Produce Prototype* focuses attention on the distributed nature of this process: three design activities are carried out by three different research groups. One can then discuss the desirability of various alternatives for managing this fit dependency.

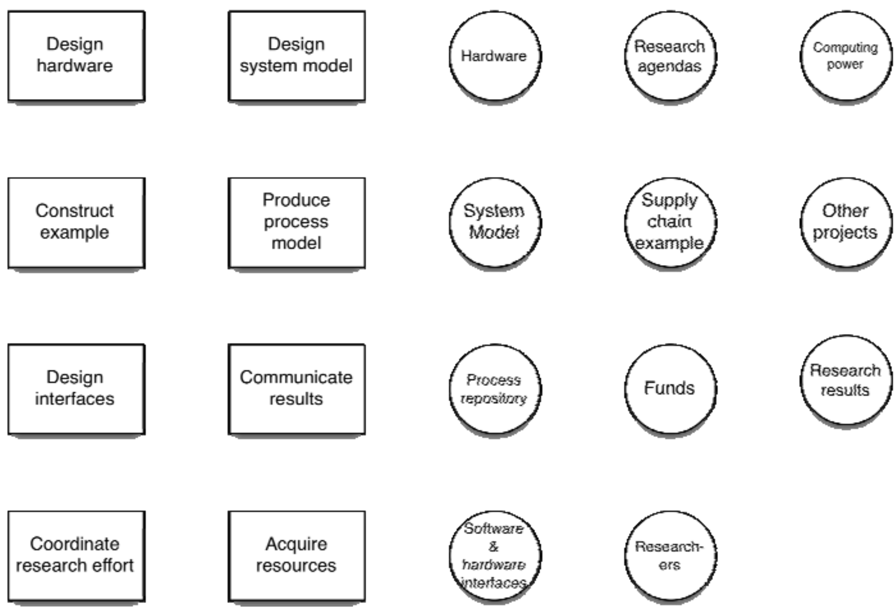


Fig. 2. CTU process fragment diagram

The claim here is not that we could only arrive at this result by using RFGA. Clearly the method described by Crowston and Osborn might have resulted in a very similar diagram. The claim is instead that RFGA gives one a systematic way to identify all relevant resource flows in the process and thus a comprehensive set of dependencies.

5 Validation

Hevner et al [9] suggest that novel design artifacts (including design methods) may need to employ relatively informal evaluation techniques including informal arguments and scenarios. I will use both these approaches to assess claims of usability and usefulness for RFGA.

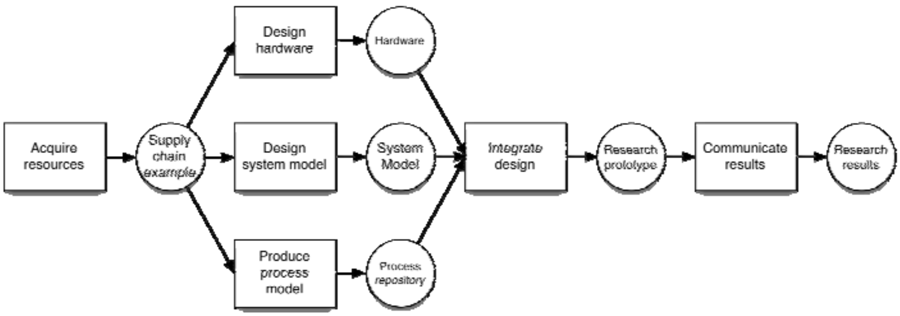


Fig. 3. CTU resource flow graph

5.1 Preliminary Assessment of Usability

Given the novelty of RFGA, the evaluation carried out assesses a weak claim of usability: the claim that the method can be used in practice to develop a dependency diagram for some actual organizational process. I evaluated this claim by carrying out a number of "design exercises," one of which was described in section 4 above. Each design exercise included the construction of a dependency diagram, which then served as the basis for an analysis of the organizational situation described. At the close of each exercise I asked the participants to provide some feedback about the method and the analysis.

In each design exercise, the conversation which resulted was in reaction to diagrams developed by me in response to a situation presented by the participant. The methodology thus was the central resource for structuring and guiding each design exercise. The insights obtained from these exercises and their reported relevance to

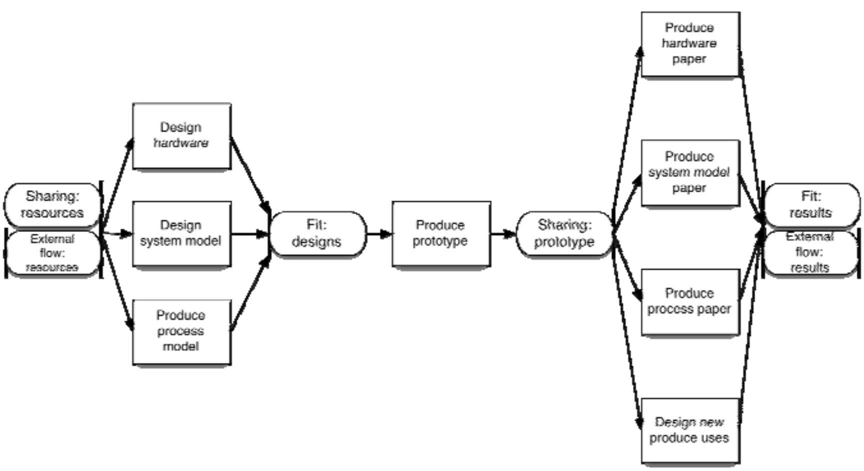


Fig. 4. Dependency diagram: DESIGN RESEARCH PROTOTYPE

the situations described by participants would seem to provide preliminary support for the claim that this methodology is usable and at least limited support that it has the potential to be useful as well [17].

In addition, the method has been used to analyze the process by which application service providers deliver services to their customers [14] and to identify coordination issues and design alternatives for knowledge management systems [15].

5.2 Preliminary Assessment of Usefulness

To assess its usefulness, I will briefly consider the potential value of RFGA relative to the Crowston and Osborn method discussed above. RFGA is intended as a complement to the Crowston and Osborn approach, specifically as an elaboration on the method of tracing resource flows in order to discover dependencies. The principle capabilities which RFGA brings to resource tracing are (1) the use of the resource flow graph to represent resource flows and (2) the iterative method by which the resource flow graph is obtained. By showing all resources and activities in relation to each other and within the context of the overall process, it becomes easier to identify flows that might have been overlooked. Further, by building up the resource flow graph from a series of fragments, one is able to begin with what may be the insights of individual process stakeholders about small fragments of the process and then gradually see how these fragments can be integrated into a larger view.

While RFGA adds some additional techniques for identifying and validating dependencies, these additional steps may not be necessary when the initial process descriptions are comprehensive or when resource flows are already well understood. In such cases it might be preferable to use the Crowston and Osborn approach to arrive at a dependency diagram directly, without first creating a resource flow graph. The additional capabilities of RFGA seem likely to be especially useful in circumstances when process descriptions are partial and inconsistent and when it is difficult to ascertain the role played by resources in the process.

6 Conclusion

This paper describes a method for developing dependency diagrams from informal process descriptions by first identifying resources and activities and the relationships among them. The approach addresses difficulties inherent in this kind of analysis including the difficulty of observing dependencies directly and the distributed nature of process knowledge. The method was evaluated by a series of design exercises and a comparison to existing methods. While preliminary assessments of the method suggest its potential as a tool for analysts, additional work is necessary to realize this potential:

RFGA must be subjected to a more rigorous test of its usability and usefulness. The method should be employed on larger projects with measurements made of project outcomes and analyst perceptions. The role played by RFGA in the broader systems development process also needs to be explored further. The method might be extended to provide more guidance for moving from a dependency diagram to system requirements such as use cases. Some preliminary work along these lines has already been carried out [18].

While we have made the case for a dependency view based on coordination theory, it is worth noting that other researchers have proposed methods for representing dependencies for this purpose [19-21]. RFGA should be compared to these methods to identify the trade-offs and potential complementarities. For example, a preliminary comparison with Dependency Network Diagrams [21] suggests that RFGA might be a useful complement in situations where system requirements are driven by a fine grained view of dependencies among individual activities as opposed to soley based on patterns of interdependence among organizational actors. Research by Reijers et al seems especially worth comparison with RFGA since it appears to apply a similar approach to the analysis of information products [22].

Acknowledgments. The author would like to thank Thomas Malone, Kevin Crowston, Brian Pentland, and Jintae Lee for their many insights into coordination theory and dependency diagrams. A special thanks to the late Charles Osborn for his insights into how to discover dependencies and analyze dependency diagrams. My thanks to Greg from CTU for his participation in the design exercise described above.

References

1. Avison, D.E., Taylor, V.: Information systems development methodologies: A classification according to problem situation. Journal of Information Technology 12, 73–81 (1997)
2. Curtis, B., Kellner, M.I., Over, J.: Process modeling. Communications of the ACM 35, 75–90 (1992)
3. Basu, A., Kumar, A.: Research commentary: Workflow management issues in e-business. Information Systems Research 13, 1–14 (2002)
4. Davenport, T.H.: Process Innovation: Reengineering Work through Information Technology. Harvard Business School Press, Boston (1993)
5. Hammer, M., Champy, J.: Reengineering the Corporation: a Manifesto for Business Revolution. HarperCollins Publishers, Inc., New York (1993)
6. Malone, T.W., Crowston, K., Lee, J., Pentland, B., Dellarocas, C., Wyner, G., Quimby, J., Osborn, C.S., Bernstein, A., Herman, G., Klein, M., O'Donnell, E.: Tools for inventing organizations: Toward a handbook of organizational processes. Management Science 45, 425–443 (1999)
7. Malone, T.W., Crowston, K.: The Interdisciplinary Study of Coordination. ACM Computing Surveys 26, 87–119 (1994)
8. Crowston, K., Osborn, C.S.: A Coordination Theory Approach to Process Description and Redesign. In: Malone, T.W., Crowston, K., Herman, G. (eds.) Organizing Business Knowledge: The MIT Process Handbook, pp. 335–370. MIT Press, Cambridge (2003)
9. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. Mis Quarterly 28, 75–105 (2004)
10. March, S.T., Smith, G.F.: Design and natural science research on information technology. Decision Support Systems 15, 251–266 (1995)
11. Yourdon, E.: Modern Structured Analysis. Yourdon Press, Englewood Cliffs (1989)
12. Larman, C.: Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. Prentice-Hall PTR, Upper Saddle River (2005)
13. Rumbaugh, J., Jacobson, I., Booch, G.: The Unified Modeling Language Reference Manual. Addison-Wesley, Boston (1999)

14. Balasubramanian, P.R., Wyner, G.M., Joglekar, N.: The Role of Coordination and Architecture in Supporting ASP Business Models. In: Thirty-Fifth Hawaii International Conference on System Sciences. Computer Society Press of the IEEE, Waikoloa (2002)
15. Iyer, B., Shankaranarayanan, G., Wyner, G.: Process Coordination Requirements: Implications for the Design of Knowledge Management Systems. Journal of Computer Information Systems 46, 1–13 (2006)
16. Zlotkin, G.: Managing Shared Resource (1995)
17. Wyner, G.M.: Let a Thousand Gardeners Prune: Cultivating the Distributed Design of Complex Organizations (2000)
18. Liu, X., Wyner, G.M.: Coordination Analysis: A Method for Deriving Use Cases from Process Dependencies. In: 4th International Conference on Design Science Research in Information Systems and Technology (DESRIST 2009), Malvern, Pennsylvania (2009)
19. Basu, A., Blanning, R.W.: A formal approach to workflow analysis. Information Systems Research 11, 17–36 (2000)
20. Kumar, A., Zhao, J.L.: Dynamic routing and operational controls in Workflow Management Systems. Management Science 45, 253–272 (1999)
21. Tillquist, J., King, J.L., Woo, C.: A representational scheme for analyzing information technology and organizational dependency. MIS Quarterly 26, 91–118 (2002)
22. Reijers, H.A., Limam, S., van der Aalst, W.M.P.: Product-based workflow design. Journal of Management Information Systems 20, 229–262 (2003)

Analyzing Web Service Choreography Specifications Using Colored Petri Nets

Enrique Caliz, Karthikeyan Umapathy,
Arturo J. Sánchez-Ruíz, and Sherif A. Elfayoumy

School of Computing, University of North Florida, 1 UNF Drive,
Jacksonville, FL 32224, USA
ecaliz@yahoo.com,
{k.umapathy,asanchez,selfayou}@unf.edu

Abstract. Enacting cross-organizational business processes requires critical support for long-running and complex interactions involving multiple participants. The Web Services Choreography Description Language (WS-CDL) aims at facilitating just that, by providing means to describe correlated message exchanges among services geared towards achieving a business goal. While WS-CDL specifications are machine-readable documents, they do not necessarily allow developers to determine—by direct inspection—whetheror not the patterns of message exchanges they stipulate do indeed describethe intended service behavior. In this research paper, we show how Colored Petri Nets (CPN) can be used to analyze WS-CDL documents in order to identify faults in the specification. We have developed a research prototype that assists in the creation of a CPN model from a given WS-CDL document. The CPN model generated is then analyzed using the formal verification environment and simulation capability provided by CPN-Tools. We provide a discussion on the analysis of an example WS-CDL document using this approach, as well as on the advantages and limitations of using CPN for analyzing WS-CDL specifications.

Keywords: Web service, WS-CDL, Service Choreography, Colored Petri Nets, ChorToNet,and Analyzing Specifications.

1 Introduction

Supporting interactions among Web services is important in the context of cross-organizational business processes as they typically involve multi-step as well as correlated message exchanges among services. Successful completion of long-running interactions depends upon multi-party conversation policies that specify "who is allowed (or expected) to send messages to whom and in what order" [1]. Conversation policies would necessitate services to establish an explicit conversation context during their initial contact and interact with each other within that context in order to attain their goal [2]. Services interpret each message exchanged in relation to the previously exchanged messages in the framing conversation. Thus, conversation policies that describe peer-to-peer interactions among participating services can be considered as state-transition models as they describe a reactive behavior of services [3].

H. Jain, A.P. Sinha, and P. Vitharana (Eds.): DESRIST 2011, LNCS 6629, pp. 412–426, 2011.

The Web Services Choreography Description Language (WS-CDL)[4] is a XML-based specification being proposed by the World Wide Web Consortium (W3C). The purpose of WS-CDL specification is to provide a systematic way to specify conversation policies that describe the ordering of message exchanges amongWeb services.The WS-CDL specification describes peer-to-peer interactions among participating services, as well asthe type of messages that need to be exchanged, in a well-defined order that may contain conditional branches and loops.

It should be noted that WS-CDL has gained little traction in practice, and W3C has not formally recommended it as a standard. However, WS-CDL specification, currently, is the best solution available for supporting complex interactions among Web services. Adequate tool support for developers is one of the key ingredients to increase adoption of WS-CDL in the industry.

Although developers can use tools such as Pi4soa [5] to create a WS-CDL document, their ability to determine—by direct inspection—whether message exchanges described in a WS-CDL document are fault-free and describe the intended service behavior, is limited. A fault in a WS-CDL document, for example, would be a missing or incorrectly specified order of message exchange, which would result in unintended service interaction behavior or the impossibility for the occurrence of an otherwise viable message exchange.Thus, faults in WS-CDL specificationscan have a detrimental effect on the successful completion of the business transactions they describe. Merely reading WS-CDL documents to determine whether they contain an appropriate order of message exchanges is not practical. Therefore, it is crucial to develop effective means to analyze and identify faults in a WS-CDL document prior to deploying it.

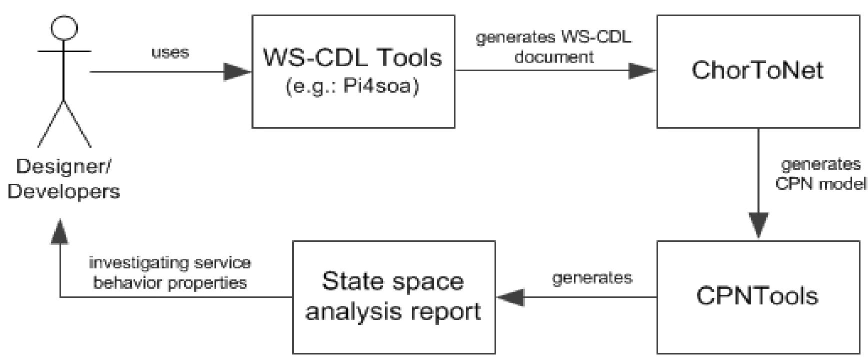


Fig. 1. Approach for investigating faults in WS-CDL documents

In this paper, we develop an approach (see Fig. 1) to analyze and search for "red flags" that signal the potential presence of faults in WS-CDL documents. Our approach takes advantage of modeling formalisms provided by Colored Petri Nets (CPN) [6] and the simulation capabilities provided by CPN-Tools [7]. Towards that end, we have developed a CPN model representation for each element of a WS-CDL specification that is relevant to the ordering of message exchanges. Since developing CPN models from WS-CDL by hand can be a tedious and error-prone task, we have developed a research prototype, dubbed ChorToNet, that takes a given WS-CDL document as input, recognizes WS-CDL elements specified in the document, and generates a CPN model that can be directly imported by CPN-Tools for further analysis. We analyze a sample WS-CDL document to demonstrate the utility of the approach described in this paper.

The rest of the paper is organized as follows: section 2 provides a background on WS-CDL and CPN; section 3 provides details on the CPN representation for WS-CDL elements we have defined; section 4 discusses details of the ChorToNet application; section 5 shows the analysis of a WS-CDL document using our approach, and—finally—in section 6, we offer concluding remarks.

2 Background

In this section, we provide an overview of WS-CDL specifications and CPN modeling. We also provide a review of related works on analyzing WS-CDL specifications.

2.1 Overview of WS-CDL

A WS-CDL specification provides a language for defining service interaction behavior rules, byproviding a complementary layer above service composition languages for coordinating service interactions. Web service composition languages such as WS-BPEL [8]describe the order of tasks performed by participating services to achieve business goals. However, they do not specify the order of message exchanges required to maintain long-running conversations among participating services.

As shown in Fig. 2, each WS-CDL document contains a *package* as its root element. Within the *package* element, details about collaborating participants and peer-to-peer interactions between them are specified. Thus, WS-CDL documents contain a static section, which provides details about collaborating participants; and a dynamic section—i.e., the*choreography* element—which provides message exchange ordering rules and constraints [9].

2.2 Overview of Colored Petri Nets

Colored Petri Nets (CPN) constitute a state-oriented modeling language for simulating and verifying whether systems have certain set of behavioral properties [10]. CPN combines strength of ordinary Petri Nets [11] with the functional programming language Standard ML [12]. Petri Nets provide the fundamental graphical notations and basic primitives such as places, transitions, and arcs. Standard ML provides the primitives for defining data types and manipulating data values.To gain a detailed understanding of CPN, please refer to [10].

A CPN model represents different possible states within the system and events that trigger system change from one state to another state [10]. A CPN model organizes system states and events into a set of modules (pages) each of which consists of a network of places, transitions, and arcs[10].Each place can be marked with tokens. Each token represents a data value, which is referred to as its color. The state of the system is modeled using tokens and places with associated token colors.

WS-CDL Elements	Description
package	package is the root element and provides attributes for namespace and author of the document
roleType	roleType represents behavior exhibited by a participant during interaction.
relationshipType	relationshipType represents commitments made by participants by identifying their respective roleTypes.
participantType	participantType provides set of roleTypes indicating behaviors that must be implemented by a participant.
channelType	channelType specifies where and how information is exchanged between participantTypes.
informationType	informationType defines type of information contained in a variable
token	token references to a piece of data in an informationType or a variable.
choreography	choreography defines behavior patterns between interacting participants.
variable	variable captures information (such as result of an exchange, observable state, and exceptions caused) about objects involved in choreography.
activity	activity describes coordinated actions to be performed by participants within a choreography.
sequence	sequence represents two or more actions being performed sequentially.
parallel	parallel represents two or more actions being performed concurrently.
choice	choice represents a structure where only one of two or more actions would be performed.
workunit	workunit describes repetitive activities and constraints for activities to performed by participants within a choreography.
assign	assign enables changing value of variables
interaction	interaction represents information exchange between collaborating participants.
perform	perform specifies another choreography either within same or external package to be performed.
silentAction	silentAction hides an action performed by a participant from other participant.
noAction	noAction indicates that a participant does not perform any action.
exceptionBlock	exceptionBlock specifies actions to be performed when a exception occurs.
finalizerBlock	finalizerBlock specifies actions to be performed when a choreography has been successfully completed.

Fig. 2. WS-CDL document structure

An event is modeled using transitions. A substitution transition is used as an abstract representation (i.e., higher level view) of a CPN page that contains a detailed description of a sub-system. The inputs and outputs of a substitution transition are interfaced with inputs and outputs of a CPN page through special places called sockets (associated with substitution transition) and ports (associated with CPN page). A port is assigned to a socket to specify the relationship between a substitution transition and a CPN page. Arcs connect places and transitions and also control the movement of tokens through the model. Transitions are enabled as tokens move from one transition to another.

Desired properties of a system under analysis can be verified by performing state space analysis on its CPN representation. As part of a state space analysis all reachable states and event occurrences are computed in order to identify system behaviors such as the absence of deadlocks, the possibility of reaching a given state, and the guaranteed delivery of a given service [10]. CPN models are used for analyzing key characteristics of a broad class of systems, including business process models [13], workflow models [14], agent systems [15], and software engineering [16]. CPN models can be constructed and analyzed using a tool suite known as CPN-Tools, developed by the CPN group at the Eindhoven University of Technology[7]. It offers a graphical user interface (GUI) front-end through which CPN models can be created, edited, and simulated. It also performs state space and performance analyses.

2.3 Related Works

Zhang et al. [17] developed an approach for modeling and validating WS-CDL specifications based on various UML diagrams. However, this approach does not provide the capability to ensure that service interaction behaviors have the intended effect at runtime. Xiangpeng et al. [18] developed a language called CDL, along with its semantics operationally defined, and apply the model-checking technique to verify correctness of a WS-CDL specification. However, in order to utilize this approach, developers would need to learn a new language, which might not be practical in some situations. Pu et al. [19] developed an approach to generate Java programs based on WS-CDL specifications, which can be executed to identify any violations in the WS-CDL specification. While the authors' approach is interesting, they did not provide details on tool support and evaluation on identifying typical WS-CDL specification faults. Foster et al. [20] developed an approach using finite state process algebra to translate WS-CDL specifications into message sequence charts. However, message sequence charts provide a static view for a specific scenario and may not be effective at representing sets of scenarios typically contained in WS-CDL. Decker et al. [21] developed an approach using Pi-calculus to perform reachability analysis for choreographies, however, the authors indicate that their approach was not effective when repeated interactions are present in the choreography. Diaz et al. [22] developed an approach to map a given WS-CDL specification into a Timed Automaton, which is then used for generating WS-BPEL skeleton structures. The authors utilize UPPAAL, a model-checker tool to verify the Timed Automaton representation against the expected system behavior. However, the authors' objective is to generate WS-BPEL documents, not necessarily to analyze and verify WS-CDL specification properties.

3 Mapping WS-CDL Elements to CPN Constructs

The objective of this research is to develop an approach (shown inFig. 1) to analyze WS-CDL documents using CPN analysis techniques. CPN models are constructed and simulated with the goal of debugging and investigating the system design properties [10]. CPN models can be simulated using CPN-Tools to investigate functional and logical behaviors of the system under different scenarios [10]. Thus, the state space analysis capability provided by CPN-Tools can be used for identifying potential faults associated with a WS-CDL document for different service behavior scenarios.

The idea of using CPN modeling technique is by no means novel, as it has been utilized by other researchers to verify other web service specifications such as WS-BPEL [23, 24] and WSCI [25, 26].

In order to analyze WS-CDL documents using CPN-Tools, first, we need to establish an appropriate mapping between WS-CDL elements and CPN constructs. In this preliminary research, to reduce the complexity associated with CPN modeling; we primarily focus on those WS-CDL elements directly involved in specifying the order in which the activities occur, as well as those associated with the exchange of information and the assigning of values to variables. In the next sections, we briefly describe CPN representations for WS-CDL elements.

3.1 *Activity* Element

*Activity*element specifiesthe ordering rules of actions performed within the *choreography* element[4]. *Activity*is a group element, i.e., contains a combination of activities; therefore,it is represented using a substitution transition. Thissubstitution transitionfires as the control token moves from one to anothersubstitution transition. A *sequenceactivity* element is represented using sequential pairing of substitution transitionwith interposing socket place. Similarly, a *parallel activity* element is represented using concurrent pairing of sequential socket places with interposing-substitution transition. In regard to a *parallel activity* element, a control token is duplicated into as many copies as there are parallel activities. Each copy of the token is passed into the substitution transitions representing each *parallel activity*. After completion of activities represented for each substitution transition, tokens are merged into one single token. A *choice activity* element is represented using concurrent pairing of sequentialsubstitution transitionswith interposingsocket place. In regard to *choice* element, the CPN-Tools randomly select a substitution transition to be executed. Fig. 3 depicts CPN model representations for *activity* elements.

3.2 *Interaction* Element

Interaction element specifies a series of message exchanges between *roleTypes*[4]. Typically, these *roleTypes* are referred to as the *fromRoleType* (the initiator of the exchange) and the *toRoleType* (the responder). Thus, exchanges between *roleTypes* can be of two types—request or response. Requests are always directed from the *fromRoleType* to the *toRoleType* and responses are from the *toRoleType* to the *fromRoleType*. *roleTypes* are represented using places, and *messages* are represented using tokens. A request message exchange is represented using a place containing a token connected to a transition which is connected to another place without a token. When the control token arrives at the transition, it is fired. The message token is then displaced from the place representing the *fromRoleType* to the place representing the *toRoleType*. Similarly, response message exchange is represented using a place without a token connected to a transition which is connected to a place with a token. Fig. 4 depicts CPN model representations for *interaction* elements.

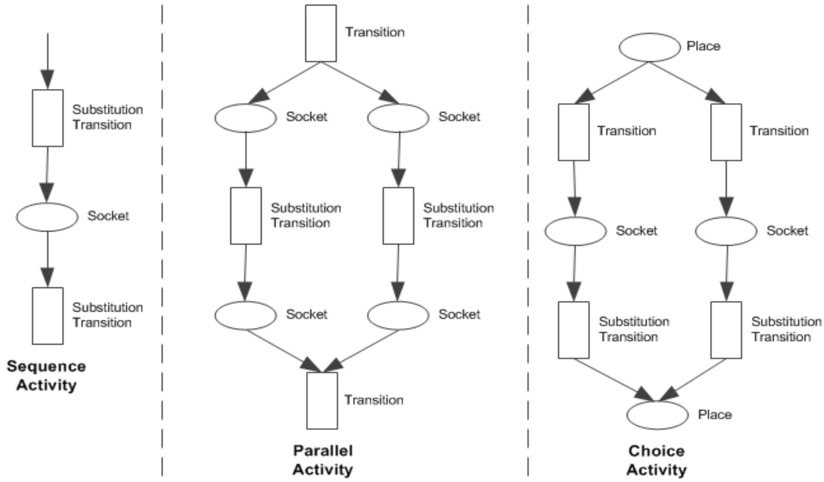


Fig. 3. CPN model representations for *activity* elements

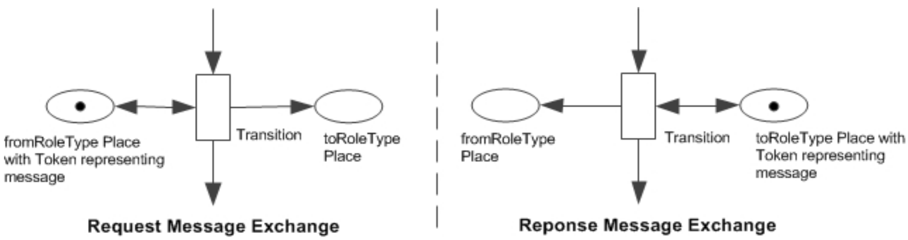


Fig. 4. CPN model representations for request and response message exchanges

3.3 *Assign* Element

Assign element is used to populate or change the value of one or more variables within a *roleType*[4].*Assign* element is represented using a CPN page wherein places representing source variables are connected to a transition, say T1. When the transition T1 is fired, the value to be assigned is placed in another place, say P1. Arc connecting T1 and P1 would contain a standard ML based expression to determine an appropriate value (as a function of the source variables) that will be placed in the P1. When the transition (say T2) connecting P1 and the place (say P2) representing the target variable is fired, then the value from P1 is inserted into the variable represented in the P2.

3.4 *Workunit* Element

Workunit element is used for specifying constraints that must be satisfied to perform certain activities and for specifying repetitive activities[4]. *Activity* elements contained within *workunit* element are represented using substitution transition. *Workunit* element also includes a guard condition, a repeat condition, and a block condition. *Workunit* element is represented using a CPN page. In the CPN page, a place, say P1 holds a token possessing a boolean variable. This variable will be true if a guard condition has been specified in which case values in places containing variables relevant to guard conditions are evaluated using the Standard ML. If guard condition was evaluated to be trueorno guard condition was specifiedthen control passes directly to the *activities* contained in the page. If guard condition is evaluated to be false, then the token in theplace(representing block attribute), say P2, is evaluated. If block condition is evaluated to be true, then the control token is redirected to the page's entry point. If block condition is false, thenpage is exited (the control token leaves the page).Upon completion of the *activities*, the value of the token in theplace (representing repeat attribute), say P3, is evaluated. If true, values in places containing variables relevant to repeat conditions are evaluated. The result of this evaluation determines if the page will be exited or re-entered from the beginning (the control token is redirected to the page's entry point).

3.5 *Choreography* Element

Choreography element specifies a pattern of collaborating behaviors between services[4]. *Choreography* element is represented using a CPN page containing representations for *interaction* and *activity* elements contained within the *choreography* element.

4 ChorToNet Application

Manually transforming a WS-CDL document into its equivalent CPN model representation can be laborious and time consuming. However, both WS-CDL and CPN models can be generated as XML-based documents, so automating the construction of one based on information gathered from the other is quite feasible. Thus, the purpose of the ChorToNet application is to generate CPN model representations of the choreography sections of WS-CDL specifications. This application takes a WS-CDL documentas input, applies the mappings discussed in the previous sections, and produces an XML document that is readable by CPN-Tools.

We initially considered using eXtensibleStylesheet Language Transformations (XSLT) to transform WS-CDL specifications into CPN model representations. XSLTs are best suited for mappings that are more of a static nature such as transforming XML documents into XHTML documents. However, generating CPN models based on WS-CDL specification is a complex transformation. Consider, for instance, the tasks of coordinating positions of places and transitions to ensure the CPN model can be visually inspected using CPN-Tools.

The ChorToNet application was developed using Java following a layered architecture, as shown in Fig. 5. It uses XmlBeans to generate Java types that represent XML schema types of WS-CDL specifications and CPN models. The input WS-CDL document is traversed, and the required information (e.g., sequence, choice, parallel, interactions, workunits, roleTypes, and variables involved in the choreography) are appropriately collected. Then this information is used by the CPNBuilder class to create the CPN model document. The CPNBuilder class uses cpn-choreography_Elements layer, which provides the functionality for transforming WS-CDL elements into CPN representations. Classes in the cpn-choreography_Elements layer use builders layer for the functionality required to construct places, arcs, and transitions (including substitution transitions). In the next section, we illustrate our approach by analyzing a sample WS-CDL document, and identifying and rectifying faults contained in it.

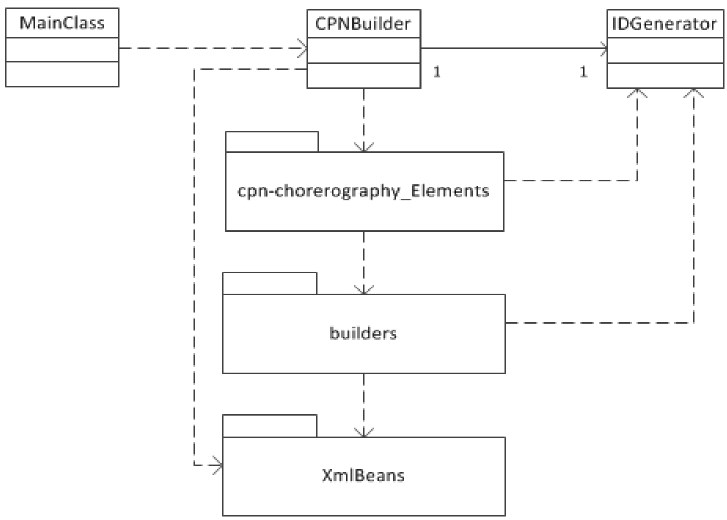


Fig. 5. ChorToNet Architecture

5 Analyzing a Sample WS-CDL Document: Intermediate Example

The approach espoused in this paper was applied to a simple problem named 'Intermediate Example,' which is found in the WS-CDL Primer developed by W3C [27]. The 'Intermediate Example' is a *choreography* that describes the sequence of message exchanges between two *roleTypes* that represent a buyer and a seller of some commodity. Thus, in this *choreography* we have two *roleTypes*: BuyerRole and SellerRole, and the variable controlling the flow of the *choreography*isbarteringDone, which is a Boolean type. At the start of the *choreography*, barteringDone is assigned the value of false. Next, an *interaction* in which there is a request exchange where the BuyerRole elicits a quote from the SellerRole. Subsequently, there is a response exchange in which the SellerRole sends a quote to the BuyerRole. The process then enters into a loop whereby the BuyerRole either accepts or rejects the quote. If the buyer accepts the quote, the barteringDone variable is set to true and the process terminates. If the quote is rejected, a quote is re-elicited. This cycle continues until the quote is accepted and barteringDone is assigned true.

Using the ChorToNet application, a CPN model representation of the WS-CDL document was generated. Subsequently, the CPN-Tools was used to generate the state space report (see Fig. 6),whose analysis allowed us to detect faults in the specification. Table 1 provides a brief explanation of the state space report properties and corresponding interpretation for analyzing WS-CDL documents. From the report, it can be noted that there is an identical number of nodes in both the state space and the strongly connected component (SCC) graph. This means that no SCC had more than one node associated with it. Thus, there was no potential for cycles. This was an unexpected behavior as the bartering process was expected to continue until the quote was accepted. Since it is possible the quote is never accepted, there should be a cycle somewhere in the state space. This indicates that there is a fault in the WS-CDL specification, which prevents the bartering process from being cyclic.

From the report (cf. Fig. 6), it can be noted that both Home marking and Dead marking is represented by node number 12. This is not unusual as it means there is a terminal state (node 12), and that state is reachable from all other states. Further analysis of the state space using CPN-Tools, showed node 12 coincided with the completion of the choreography. Next we looked at the dead transitions. Investigation of these dead transitions indicated the *activity* within the *workunit* was never carried out. After the quote was received by the BuyerRole, the process terminated. In other words, no bartering takes place. Clearly, this is not the desired service behavior. The fact that process terminates after the BuyerRole receives the quote indicates that

```
CPN-Tools state space report
       State Space
              Nodes: 12
              Arcs: 11
              Secs: 0
              Status: Full
       Scc Graph
              Nodes: 12
              Arcs: 11
              Secs: 0
       Home Markings
              [12]
       Dead Markings
              [12]
       Dead Transition Instances
              AssignID135 T145
              AssignID135 T146
              AssignID135 T147
              ChoiceID271 T281
              ChoiceID271 T295
              Interaction_QuoteAcceptID110 T126
              Interaction_QuoteReelicitationID212 T228
              Interaction_QuoteReelicitationID212 T246
              WorkUnit_WhileBarteringIsNotFinishedID309 T381
              WorkUnit_WhileBarteringIsNotFinishedID309 T382
              WorkUnit_WhileBarteringIsNotFinishedID309 T383
              WorkUnit_WhileBarteringIsNotFinishedID309 T385
              WorkUnit_WhileBarteringIsNotFinishedID309 T387
              WorkUnit_WhileBarteringIsNotFinishedID309 T388
       Live Transition Instances
              None
```

Fig. 6. State space report for intermediate example

the error was in the *workunit*'s guard expression. Currently, the guard condition is set to be true. Therefore, *activity* elements contained within *workunit* would be executed only when barteringDone has value true. However, initial value of barteringDone is false. Consequently, the guard condition is never satisfied. As a result, the *workunit*'s*activity* is never carried out. The report also states that there were no live transitions. This was to be expected, as there was a dead marking.

Table 1. State space report properties and their interpretations for WS-CDL documents

State space report property	Meaning for CPN model	Interpretation for WS-CDL analysis
State space	It indicates the number of reachable states (nodes) and state changes (arcs). A full state space indicates that there is a node for each state and an arc for each state change. "Secs" indicate the number of seconds taken for constructing the full state space.	Reachability of a state change can be used to verify certain service behaviors. For example, state change indicating service A sending request message to service B.
SCC graph	It indicates mutually reachable nodes in the state space, i.e., there exists a path from a given node to any other in the component.	If the number of nodes in the state space and the number of nodes in the SCC graph are equal, then each SCC is comprised of exactly one node and thus there are no cycles. If the number of SCCs is less than the number of nodes, then one or more cycles are present as at least one SCC must have more than one node. In this case, the nodes in those multi-node SCCs must be analyzed to determine if the cycle(s) corresponded with what is expected.
Home markings	A marking reachable from any reachable markings. A marking represents a node in the state space.	The presence of more than one home marking implies the presence of infinite cycles.
Dead marking	A marking with no enabled transitions represents a terminating state.	Presence of dead marking indicates completion of a process or reaching a state from where it will be unable to continue. Presence of no dead markings indicates that process will never terminate.
Dead transition instances	Transitions with no reachable marking that are enabled. Dead transitions indicate a part of the model that will never be activated.	Presence of dead transitions may indicate that a certain interaction can never occur, i.e., associated interaction may have no effect on service behavior.
Live transition instances	Transition that can be enabled once from any reachable marking.	Presence of live transition indicates non-terminating loops.

5.1 Modified Intermediate Example

A modification was introduced into the 'Intermediate Example' WS-CDL document. Specifically, the guard expression of the *workunit* was changed from true to false. This was done with the intention of removing the fault discovered in the previous section. The state space report associated with the modification is shown in Fig. 7. From the report, it can be noted the size of the state space has increased. This was expected due to the loop associated with bartering. It can also be noted that there are fewer nodes in the SCC graph (21 nodes) than in the state space (31 nodes). This means at least one of the SCCs contains two or more nodes from the state space. Further investigation of the nodes using the CPN-Tools indicates that a SCC was associated with the *workunit* or with an *activity* contained within the *workunit*. This was an expected service behavior. This SCC is an expression of the fact that bartering could go on forever.

From the report, it can be also noted there is one dead transition. Inspection of the CPN model indicates this transition is activated only if there is some possibility that the *workunit* could block waiting for the guard condition to become true. In this case blocking was not specified in the WS-CDL document; therefore, blocking would not occur. This means the dead transition T385 should never become enabled. Again, there is no transition that is always enabled. For this version of the choreography, we found no "red flags" that can signal faults. The potentially infinite loop detected by CPN-Tools is an expected service behavior, as this means it is possible for the bartering process to continue indefinitely.

```
CPN Tools state space report
      State Space
            Nodes: 31
            Arcs: 31
            Secs: 1
            Status: Full
      Scc Graph
            Nodes: 21
            Arcs: 20
            Secs: 0
      Home Markings
            [31]
      Dead Markings
            [31]
      Dead Transition Instances
            WorkUnit_WhileBarteringIsNotFinishedID309 T385
      Live Transition Instances
            None
```

Fig. 7. State space report for modified intermediate example

6 Discussion and Future Work

In this paper, we present a preliminary report of a design science research that addresses the problem of analyzing WS-CDL documents, which specify expected service behaviors that are critical for successful completion of complex business transactions using Web service-based solutions. To analyze and identify faults in WS-CDL documents, we introduce an approach that takes advantage of CPN modeling technique and CPN-Tools. We developed a mapping between WS-CDL elements and CPN model representations. The developed tool ChorToNet automates the generation of CPN models from a given WS-CDL document. Developers then can import this CPN model document into CPN-Tools for simulating and generating state space reports to identify faults in the WS-CDL document. We have also provided guidelines for analyzing WS-CDL documents based on state space report properties. We have demonstrated the utility of the approach by identifying and correcting faults with 'Intermediate Example' found in the WS-CDL Primer specification developed by W3C.

The contribution of this paper apart from the approach and ChorToNet application, is that by analyzing 'Intermediate Example,' we have shown that even the creator of the WS-CDL specification standard, W3C, has introduced errors in developing example choreography. This indicates how easy it is to introduce design flaws when developing WS-CDL documents, thereby, demonstrating importance of this research.

Using CPN technique, however, presents a major disadvantage. The limitation of the approach adopted in this paper is that developers have to interpret state space report properties and need to have knowledge of standard ML for further investigation of report properties. These are not common traits of a Web service developer. Thus, in order to make this approach practical, we would have to automate the process of simulating and generating state space reports using CPN-Tools, interpret report properties, and annotate WS-CDL document with identified possible faults. Achieving the above is an objective of our future work. We recognize that we have critiqued certain related works that exhibit similar limitations. However, we argue that unlike other modeling formalisms, CPN provides a dual advantage: it portrays process logic and message exchange behaviors, and also provides the strong capability for analyzing and verifying design expectations of Web service specifications in the form of state space reports and CPN-Tools.

In conclusion, this research contributes to the Web servicedomain, by extending application of CPN modeling technique to analyze Web service choreography specifications. We have shown that our approach can detect possible faults in WS-CDL specifications. Overcoming the limitations and completely automating the approach presented would mean reduced costs and time associated with developing and verifying Web service choreography specifications before deploying it.

References

1. Hohpe, G.: Let's Have a Conversation. IEEE Internet Computing 11, 78–81 (2007)
2. Hanson, J.E., Nandi, P., Kumaran, S.: Conversation support for Business Process Integration. In: IEEE International Enterprise Distributed Object Computing Conference (EDOC), pp. 65–74. IEEE Computer Society, Los Alamitos (2002)

3. Umapathy, K.: From Service Conversation Models to WS-CDL. In: Americas Conference on Information Systems (AMCIS). Association for Information Systems (AIS), Atlanta (2009)
4. W3C, `http://www.w3.org/TR/ws-cdl-10/`
5. Pi4 Technologies Foundation, `http://pi4soa.sourceforge.net/`
6. Jensen, K.: Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use. Springer, Heidelberg (1995)
7. CPN Group, University of Aarhus, `http://wiki.daimi.au.dk/cpntools/cpntools.wiki`
8. OASIS, `http://www.oasis-open.org/committees/download.php/23964/wsbpel-v2.0-primer.htm`
9. Fredlund, L.-Å.: Implementing WS-CDL. In: Spanish Workshop on Web Technologies. University of Santiago de Compostela, Santiago de Compostela, Spain (2006)
10. Jensen, K., Kristensen, L.M., Wells, L.: Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems. International Journal on Software Tools for Technology Transfer 9, 213–254 (2007)
11. Peterson, J.L.: Petri Net Theory and the Modeling of Systems. Prentice Hall PTR, Upper Saddle River (1981)
12. Milner, R., Tofte, M., Harper, R., MacQueen, D.: The Definition of Standard ML, Revised edition. The MIT Press, Cambridge (1997)
13. Liu, R., Bhattacharya, K., Wu, F.: Modeling Business Contexture and Behavior Using Business Artifacts. In: Krogstie, J., Opdahl, A., Sindre, G. (eds.) CAiSE 2007 and WES 2007. LNCS, vol. 4495, pp. 324–339. Springer, Heidelberg (2007)
14. Ha, S., Suh, H.-W.: A timed colored Petri nets modeling for dynamic workflow in product development process. Computers in Industry 59, 193–209 (2008)
15. Cost, R., Chen, Y., Finin, T., Labrou, Y., Peng, Y.: Using Colored Petri Nets for Conversation Modeling. In: Dignum, F., Greaves, M. (eds.) Issues in Agent Communication. LNCS, vol. 1916, pp. 178–192. Springer, Heidelberg (2000)
16. Denaro, G., Pezzè, M.: Petri Nets and Software Engineering. In: Desel, J., Reisig, W., Rozenberg, G. (eds.) Lectures on Concurrency and Petri Nets. LNCS, vol. 3098, pp. 439–466. Springer, Heidelberg (2004)
17. Zhang, P., Li, B., Muccini, H., Zhou, Y., Sun, M.: Data-Enriched Modeling and Verification of WS-CDL Based on UML Models. In: IEEE International Conference on Web Services, pp. 752–753. IEEE, Los Alamitos (2008)
18. Xiangpeng, Z., Hongli, Y., Chao, C., Xiwu, D., Zongyan, Q.: Verification of WS-CDL Choreography. In: Asian Working Conference on Verified Software. UNU-IIST, Macao SAR, China (2006)
19. Pu, G., Shi, J., Wang, Z., Jin, L., Liu, J., He, J.: The Validation and Verification of WSCDL. In: Asia-Pacific Software Engineering Conference, pp. 81–88. IEEE, Los Alamitos (2007)
20. Foster, H., Uchitel, S., Magee, J., Kramer, J.: Model-Based Analysis of Obligations in Web Service Choreography. In: International Conference on Internet and Web Applications and Services. IEEE Computer Society, Los Alamitos (2006)
21. Decker, G., Zaha, J., Dumas, M.: Execution Semantics for Service Choreographies. In: Bravetti, M., Núñez, M., Zavattaro, G. (eds.) WS-FM 2006. LNCS, vol. 4184, pp. 163–177. Springer, Heidelberg (2006)
22. Díaz, G., Cambronero, M.E., Pardo, J.J., Valero, V., Cuartero, F.: Model Checking Techniques applied to the design of Web Services. CLEI Electronic Journal 10, 2–14 (2007)

23. Yang, Y., Tan, Q., Xiao, Y., Yu, J., Liu, F.: Exploiting hierarchical CP-nets to increase the reliability of Web services workflow. In: International Symposium on Applications and the Internet (SAINT), p. 7. IEEE, Los Alamitos (2006)
24. Li, Y., Ye, L., Dague, P., Melliti, T.: A Decentralized Model-Based Diagnosis for BPEL Services. In: IEEE International Conference on Tools with Artificial Intelligence, pp. 609–616. IEEE Computer Society, Los Alamitos (2009)
25. Huang, Y., Xu, C., Wang, H., Yunn, X., Zhu, J., Zhu, C.: Formalizing Web Service Choreography Interface. In: International Conference on Advanced Information Networking and Applications Workshops, pp. 576–581. IEEE Computer Society, Los Alamitos (2007)
26. Deng, X., Lin, Z., Cheng, W., Xiao, R., Li, L., Fang, L.: Modeling and verifying web service composition using colored petri nets based on WSCI. In: IEEE International Conference on Industrial Engineering and Engineering Management, pp. 1863–1867. IEEE, Los Alamitos (2007)
27. W3C, `http://www.w3.org/TR/ws-cdl-10-primer/`

Neuroscience in Design-Oriented Research: Exploring New Potentials

Jan vom Brocke[1], René Riedl[2], and Pierre-Majorique Léger[3]

[1] University of Liechtenstein,
Fürst-Franz-Josef-Strasse 22, 9490 Vaduz, Liechtenstein
jan.vom.brocke@uni.li
[2] University of Linz,
Altenbergerstr. 69, 4040 Linz, Austria
rene.riedl@jku.at
[3] HEC Montréal,
Chemin de la Côte-Sainte-Catherine, 3000 Montréal, Canada
pierre-majorique.leger@hec.ca

Abstract. Design-oriented research has evolved as a major research paradigm in the academic discipline of information systems (IS) aiming at the design of innovative and useful IT artifacts such as methods, models, constructs, and instantiations. With the concept of "user-perception" at the core of this approach, it appears promising to explore the potentials of neuroscience in design-oriented research that allow for measuring physiological effects of people interfering with artifacts. In this paper, we discuss fields of application concerning both the design and evaluation of artifacts. However, we also argue that neuroscience, despite its value for design-oriented IS research, should complement rather than substitute traditional research approaches and that results require thorough interpretation. We report on a first study that triangulates quantitative and neuroscientific data in the area of enterprise resource planning systems and indicate directions for future research.

Keywords: design science research, design theory, brain, neuroscience, fMRI.

1 Introduction

Design-oriented research concerns the design process of IT artifacts, for example, constructs, models, methods, and instantiations [23, 30, 34, 51]. The design of conceptual models at an enterprise-wide scale may serve as an example highly relevant in practice. While the origins of conceptual modeling can be traced back to software engineering, there are several additional purposes of conceptual modeling essentially referring to the different areas that are objects of modeling. Among these are data modeling [4, 43], knowledge modeling [10, 32], business process modeling [3, 48], or – in its broadest sense – enterprise modeling [17, 47].

The design of conceptual models comes along with questions associated with human perception. Examples are: Which modeling language/technique is the most

H. Jain, A.P. Sinha, and P. Vitharana (Eds.): DESRIST 2011, LNCS 6629, pp. 427–439, 2011.

appropriate? Which is the right level of detail? How to safeguard the comprehensibility of process models developed for people from different cultural backgrounds? And, to what extent do people perceive these models as useful for their individual work at all? Also, recent studies have increasingly accounted for the role of user perception in conceptual modeling, putting constructs such as "model understandability" at the centre of interest [31, 36]. Moreover, other disciplines also stress the importance of user perception when designing IT artifacts. Examples are requirements engineering [42, 46], usability engineering [33] as well as human-computer interaction [8].

In design-oriented research, traditional qualitative and quantitative approaches such as interviews and surveys have been suggested as tools to assess the appropriateness of artifacts based on conscious perceptions of users [23]. However, the use of such methods may be associated with limitations (e.g., [38]) since interview and survey data, for example, are often subjective and possibly influenced by hidden intentions of the informants. Hence, the reliability of such evaluations is a topic of major interest in design-oriented research. In this regard, neuroscience may offer promising tools for design-oriented research as neurobiological data is considered more reliable than traditional data sources in specific research situations [15, 16, 38]. However, neuroscience is a new approach in design-oriented research [29, 40] and there is a lack of knowledge on how to make use of the potentials neuroscience may offer.

This paper intends to explore the use of neuroscientific tools in design-oriented research. For this purpose, a brief introduction and classification of tools is given in the next chapter. We then identify important areas in design-oriented research that may benefit from the use of these tools. In order to illustrate these potentials, we report on a current Neuro-Information-Systems (NeuroIS) study in the field of process and interface design in Enterprise Resource Planning (ERP) systems. We discuss our results and conclude with a brief summary and an outlook on future research opportunities.

2 A Review on Tools from Neuroscience

This section reviews neuroscience tools on the basis of already existing overviews (see [38]). In particular, we draw upon the following papers: [7, 24, 25, 29]. The purpose is to give ground for discussing the broad spectrum of tools available for design-oriented research rather than discussing each tool extensively (see Figure 1). In this paper, we focus on both brain imaging-tools and psycho-physiological tools as they mark two types of measurement strategies.

Brain Imaging Tools

We start with brain imaging tools as they are the most striking and popular neuroscientific tools at the moment [38]. In Table 1, we briefly describe four imaging methods which we consider to be relevant for design-oriented research: electroencephalography (EEG), functional magnetic resonance imaging (fMRI), positron emission tomography (PET), and magnetoencephalography (MEG).

Table 1. Selected Brain Imaging Tools (adapted from [38])

Type	Description	Pros	Cons	Examples
EEG	Brain imaging tool that measures voltage fluctuations on the scalp that result from changes in membrane conductivity elicited by synaptic activity and intrinsic membrane processes.	High temporal resolution (i.e., milliseconds and below), and this can easily detect the time course of neural activity.	Limited spatial resolution, an infinite number of source configurations can generate identical potentials on the scalp (inverse problem).	One study [44], for example, used EEG to develop and evaluate a brain-computer interface.
fMRI	Brain imaging tool that tracks the blood flow in the brain using changes in magnetic properties due to blood oxygenation (the so-called BOLD signal). Simultaneous direct recording of neural processing and fMRI responses shows that the BOLD signal reflects input to neurons and their processing.	Provides a much better spatial resolution (a few millimeters) compared to EEG.	Temporal resolution is poorer (a few seconds) than with EEG.	One study [39], for example, used fMRI to study brain activation differences between men and women during the processing of trustworthy and untrustworthy eBay offers.
PET	The detected distribution information regarding metabolism or brain perfusion allows for brain activation inferences which can then be visualized in tomograms.	Spatial resolution is relatively high (a few millimeters).	Temporal resolution is low (several minutes to fractions of an hour). Because radioactive tracers are used, the application to healthy test persons is restricted.	One study [22], for example, used PET to investigate learning and automation processes while playing the video game Tetris.
MEG	This tool is sensitive to changes of magnetic fields that are induced by the electrical brain activity.	Temporal resolution can be compared to that of the EEG. In contrast to the EEG, MEG is also able to depict activity in deeper brain structures	The inverse problem also applies to MEG.	One study [2], for example, used MEG to study the brain mechanisms underlying shopping behavior.

Psycho-Physiological Measurement

In accordance with the existing NeuroIS literature (e.g., [16, 38]), we include psycho-physiological measurement tools within the group of neuroscience tools. Although these tools do not directly measure brain activity, the captured indicators are closely related to the nervous system. Psycho-physiological tools are probably the oldest and simplest techniques for measuring somatic states [7]. In addition, they are relatively easy to apply making them an attractive alternative to more sophisticated brain imaging tools such as fMRI. The following section gives a brief overview of important

Table 2. Selected Psycho-physiological Tools (adapted from [38])

Type	Description and example
EKG	Heart rate (often measured by EKG) can be used to detect cognitive attention because the heart rate changes as cognitive attention is directed to a situation [6].
EMG	Facial EMG can be used to measure the emotional response by attaching sensors to different parts of the face. Increased activity in the zygomatic muscle group (the smile muscle near the mouth) has been linked to positive emotion and increased activity in the corrugator muscle group (the frown muscle near the eye) has been linked to negative emotion [6].
SCR	Skin conductance response (SCR) – essentially sweating – is an indicator of arousal; the more aroused an individual becomes, the more he orshe sweats, regardless of whether the arousal is positive or negative [11]. Thus, SCR can be combined with facial EMG to understand the direction and strength of emotion.

psycho-physiological tools, namely heart rate, facial electromyography (EMG), and skin conductance response (SCR).

Psycho-physiological responses occur near instantaneously while other neurological responses (e.g., the BOLD signal) can require several seconds to occur. Bearing this in mind, psycho-physiological tools are considered "highly appropriate" for the investigation of a number of IS research questions, especially in human-computer interaction studies [40].

Apart from brain imaging tools and psycho-physiological measurement tools, further techniques are available, including brain lesions, transcranial magnetic stimulation (TMS), diffusion tensor imaging (DTI), and near-infrared spectroscopy (NIRS). These techniques are discussed in more detail elsewhere (e.g., [38]). However, brain imaging on the one hand and psycho-physiological measurement tools on the other, may give a good foundation to discuss the applicability of neuroscientific tools in design-oriented research.

3 Leveraging Neuroscience for Design-Oriented Research

Design-oriented IS research concerns the design and evaluation of IT artifacts that are both novel and purposeful [23, 34]. The outcomes of design research are artifacts along with insights regarding their applicability and usefulness in certain application contexts [41]. Neuroscientific tools may be leveraged for both designing and evaluating artifacts as we will now discuss in more detail.

We start with the evaluation of artifacts, as the potentials of neuroscience are most obvious here. At a basic level, the evaluation of an artifact allows the researcher to make statements about its usefulness and ease of use. Fellow researchers have proposed the use of traditional qualitative and quantitative approaches such as case study research and simulations for the evaluation of artifacts [23]. Not only in design-oriented research may the use of such methods be associated with limitations though (e.g., [15, 38]). Both interview and survey data, for example, are often subjective as they might be influenced by the hidden intentions of the informants. In this regard, measurement methods from neuroscience may offer additional approaches to the evaluation because they make the assessment of both cognitive and affective effects of IT artifacts on individual recipients possible. Past PET studies, for instance, measured cognitive load (e.g., [22]), and fMRI was used to identify specific brain regions that are associated with cognitive conflict (e.g., [5]) such as the anterior cingulate cortex (see also chapter 2). Hence, neuroscience could be used to further investigate various phenomena related to the perception of artifacts.

One may argue that the costs associated with neurological measurements are high and the external validity of research results is limited because individuals, in case of fMRI, need to lie in a fixed position in the scanning machine. On the one hand, however, it can be expected that technological progress will enable cheaper and more mobile measurements in the future. On the other hand, it is remarkable that – considering the psycho-physiological measurement – some ”lightweight“ measurement methods (e.g., galvanic skin response, pupil behavior, heart rate) already exist, and these methods can complement the more sophisticated techniques such as fMRI. Such “lightweight” techniques make it possible to collect neurophysiological data related to the use of artifacts not only in experimental settings, but also in the professional work environment.

Apart from engaging in specific neuroscientific experiments, established findings from earlier neuroscientific research may also well be used in design-oriented research. Such findings may inform the design of an artifact and thus contribute to its grounding. Indeed, there are neuroscientific findings, most notably related to the perception of artifacts (e.g., the above findings regarding cognitive load and cognitive conflict), that may inform design research. The selection process of modeling languages, for example, may be grounded in knowledge about the information processing capacity of the recipients (e.g., regarding the processing of objects, numbers and other characters). Works on different cognitive styles could also be used in order to enable the refinement of models in a rather multi-perspective way. Similarly, studies on the rational and creative cognitive performance could provide a valuable basis for the design of models. In this context, Riedl [37], for example, highlights the question to what extent the cognitive style of people may have an influence on the choice of either object-oriented or flow-oriented languages. A wide range of similar questions can be studied accordingly using neuroscientific theories, methods, and tools.

The evaluation of artifacts by means of neuroscientific methods may further elicit new knowledge on design processes. Potential findings include cause and effect relationships between design decisions and affective impact on users. Hence, neuroscience can also be used further developing the theoretical basis available for grounding the design of artifacts. On the one hand, this relates to design theories [21, 49] as a specific type of theory that provides normative statements about typical design processes (i.e. ”how to do something,” [20, p. 628]). On the other hand, this also holds true for other theory types (e.g., the Technology Acceptance Model, TAM). Here, it again appears particularly promising to combine traditional research approaches with neuroscientific methods. Recent NeuroIS papers already discussed (i) the theoretical insights that neuroscience offers for TAM research (e.g., the X- and C-systems concept which is associated with unconscious and conscious information processing) and (ii) the brain areas associated with the neural implementation of the constructs underlying the TAM [12, 14].

In addition to grounding a specific artifacts design, neurscientific insight may also help organizing design processes as such. In this regard, it will be interesting to see how earlier phases of the design process can be studied by means of neuroscientific methods, for instance, the creative development and collaborative discussion of potential solutions [23, p. 76]. Drawing from neuroscience, there is in fact substantial research on the identification of brain structures that are relevant to the neural implementation of creativity. For example, Dietrich [13] has proposed that the role of the prefrontal cortex in creativity is threefold. First, one has to become conscious of a novel thought in order to evaluate based on past knowledge and its appropriateness (consciousness is associated with activity in the prefrontal cortex). Second, a novel thought is only the first step in the creativity process. Once the thought has occurred, "the prefrontal cortex can bring to bear the full arsenal of higher cognitive functions to the problem, including central executive processes such as directing and sustaining attention, retrieving relevant memories, buffering that information and ordering it in space—time, as well as thinking abstractly and considering impact and appropriateness" [13, p. 1015]. Third, the prefrontal cortex is also responsible for the implementation of the expression of the insight. Design-oriented IS researchers could use these and similar insights from the brain sciences in order to advance IS theorizing on the role of creativity in engineering initiatives (e.g. software engineering or business processing reengineering). Such learning could, for example, inform the organization and management of creativity-dependent tasks in design-oriented research.

In summary, there are specific opportunities for design-oriented research to leverage prior achievements in the field of neuroscience. Specifically, neuroscience allows researchers to assess user engagement with artifacts based on the measurement of physiological responses. In addition to visible behavior and decisions, hidden effects (and emotional effects in particular) can be assessed. This may help us to better understand how users perceive IT artifacts and may also shed light on factors impacting on processes of creative problem solving and innovation. At this, we do not consider neuroscientific measurement superior to other measurement approaches per se; we rather see a great potential in triangulating with the analysis of qualitative and quantitative data gathered through more traditional data collection methods, for instance, interviews or surveys. In order to illustrate this approach, we present an example by a recent study carried out in the field of ERP system design. It extends conventional ERP usage data by a rich set of neuroscientific data and therefore makes possible deepened insights into the design of ERP systems and processes.

4 Triangulating with Neuroscientific Data: The Example of ERPsim

One example on how to triangulate with neuroscientific data in design-oriented research is the ERPsim project. ERPsim is a simulation technology developed at HEC Montréal [26, 27] that allows for simulations in realistic collaboration scenarios through the use of a real-life ERP system. One key characteristic of ERPsim is that all decisions made by the participants must be entered into the ERP system, and, in order to make those decisions, all of the information required must be extracted from standard reports provided by the ERP system. As such, one can think of ERPsim as a flight simulator for an ERP system where end-users are flying a real corporate information system in a virtual business environment. More than 100 universities worldwide and numerous Fortune 1000 organizations are using ERPsim to train end-users in order to better understand the value of enterprise systems (more information is available at *http://erpsim.hec.ca*).

In recent studies, ERPsim has been used to triangulate with neurophysiological data in order to learn about ERP-system and process design. In more detail, ERPsim is continuously used to collect neurophysiological data. For this, subjects are immersed in a realistic business situation in which end-users are using an ERP system to make decisions and to resolve complex business problems. The specific approach for data triangulation is described in fig. 1.

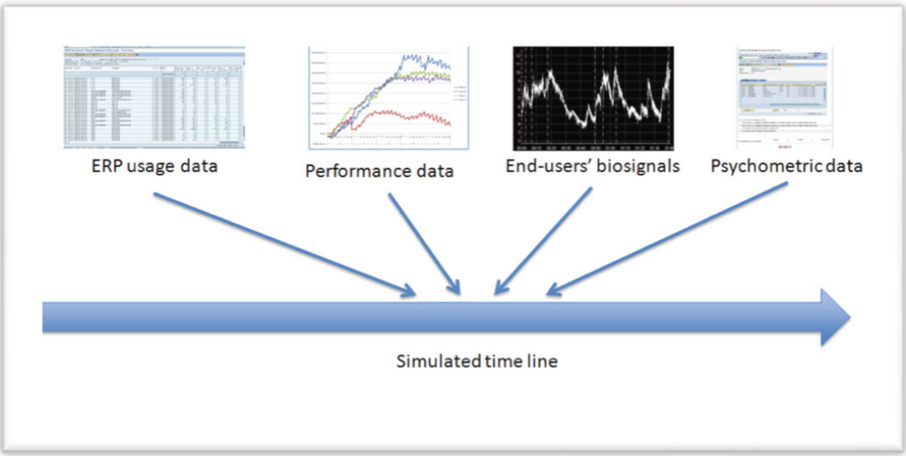


Fig. 1. Capturing a rich data set with ERPsim

As shown in Figure 1, a rich set of data is captured on users working with the ERPsim platform. In the following, we use this example in order to (i) illustrate a potential experimental setting and (ii) demonstrate some insights to be gained from triangulating with neuroscientific data in the research context of ERPsim.

Experimental Setting

The site for collecting neuroscientific data was set up as an extension of the infrastructure already existing with ERPsim. ERP systems, such as SAP, log all transactions performed by every user. This data details the sequence of all reports and transactions that each end-user executes in the system. Implementing functionality of a simulation game ERPsim makes it further possible to capture operational and financial performances achieved by the subjects performing ERP-related tasks within the simulation game.

This usage and performance data can be triangulated against neurophysiological data. With the end of a neurophysiological amplifier and appropriate data acquisition software, it is possible to capture end-users' biosignals, such as electrodermal activity

(EDA), electrocardiogram (ECG), facial electromyography (EMG), and electroencephalogram (EEG). Development is currently under way of automatically synchronising the neurophysiology data along with the ERP usage and performance data. Every simulation day, ERPsim sends a TTL (Transistor-Transistor Logic) signal to the data acquisition software in order to mark the passing of time the simulation in the neurophysiological data.

That way, the psychometric measurement tools can be used to complement the other empirical data. It particularly allows for insights into the affective effects on a user working with SAPsim in specific situations. For example, Léger et al. [28] used the SAM (Self-Assessment-Manikin) scale to obtain a self-perceived evaluation related to the emotion (valence, arousal, dominance) experienced by the subject during the experiment.

By mapping these different data sources (clickstream, survey, and biosignals) on the same timeline, it becomes evident to obtain a rich longitudinal dataset which includes the end-user's psycho-physiological reactions during the ERP experience, the self-perceived beliefs and attitudes related to this interaction, as well as a detailed record of his or her actions and decisions executed in the ERP system during the experiment together with the results achieved in the simulation game.

Exemplary Results

The triangulation with neuroscientific data in the ERPsim project allows for both evaluating ERPsim as an IT artifact and using ERPsim to investigate selected phenomena of systems design. As to the first, the data gathered is used in order to improve user interfaces and workflows implemented in the ERP system. One such example is to analyze peaks of physical arousal (measured, e. g., by heart rate or skin response) and relate those to certain stages in the process of using ERPsim. As to the latter, ERPsim is already used in order to investigate specific phenomena related to the design and perception of IT artifacts at a more general level. Current research on the concept of cognitive absorption (CA) may serve as an example:

The construct of CA corresponds to a state of deep involvement under a software program and has theoretical roots in the concept of absorption [45], the notion of flow [9] and the notion of cognitive engagement [50]. CA has widely been studied over the past decade in the IT literature using psychometric instruments developed by Agarwal and Karahanna [1]. The paradox of measuring CA with psychometric tools requires that a subject be asked to self-evaluate the level of absorption over several Likert scale items. Obviously, such an approach implies the subject to be taken out of his or her CA state in order to answer this survey.

Among the possible neurophysiological correlates of this construct (and specifically the focused attention dimension), we are currently investigating an EEG based engagement index (EI) developed by Pope et al. [35] and Freeman et al. [18]. According to Freeman et al. [19], an engagement is a state of high alertness to task relevant stimuli. Freeman et al. [18] argues that increases in beta activity are associated with a higher level of engagement related to a task, and that increases in alpha and/or theta activity would reflect less alertness and task engagement due to decreased information processing. Freeman et al. [19] suggests that EI be measured with the array of percent power on three bandwidths (theta: 4–7 Hz, alpha: 8–12 Hz and beta: 13–30 Hz) with a four electrode montage sites (03, 04, F3, and F4); specifically if IE is computed when dividing beta by the sum of alpha and theta. An ongoing research conducted in collaboration with researchers from University of Arkansas is currently investigating the relationship between this EEG-based EI and psychometric-based CA. Using ERPsim, EEG data and self-reported CA were captured from 36 novices and experts who were monitored while interacting with SAP. Results are currently being analyzed. One objective is to predict perceptions of CA based on objective psycho-physiological measurement.

The lessons learnt from the ERPsim project suggest that triangulating with neurophysiological data can be used to enhance design-oriented research in different ways. It is noteworthy that these results can already be gained using "lightweight" measurement tools. In addition, data is collected in settings that closely resemble real-life scenarios. Hence, both costs are comparably low and results show comparably high external validity. It will be interesting to see what further research questions can be addressed using ERPsim as an innovative research platform collecting a rich set of data pertaining to the behaviors and emotions of users while interacting with IT.

5 Discussion

An intriguing possibility of neuroscience is to investigate hidden factors like emotions that are related to the perception of artifacts. Such knowledge becomes critical when IS researchers investigate the mechanisms underlying the adoption as well as the design of IT artifacts. While we see great potential in using neuroscience in design-oriented research, we do not deem neurophysiological measurement superior (or more objective) to other methods per se. On the contrary we perceive neuroscience rather an opportunity to complement existing methods.

That said, it appears vital to learn further about specific strengths and limitations of neuroscience in design-oriented research. Former research has identified major weaknesses of neurophysiological tools, including cost and accessibility, the artificial setting, the labor-intensive data extraction and analysis, measurement issues, manipulation, and ethics as well as the difficulty in interpreting neurophysiological results [16]. In the following, we indicate some major issues we see particularly related to design-oriented research.

In principle, researchers should bear in mind that neuroscientific measurement refers to individual subjects only (usually about 15 to 20 subjects in fMRI studies). The utility of a wide range of artifacts, however, is also the result of the social context in which they are applied (cf. [30]). Neuroscience mainly contributes a better understanding of IT use at an individual level, while phenomena on a group and organizational level require further interpretation and observation.

In addition, the data captured in neuroscientific studies requires thorough interpretation. While great achievements have been made in identifying neural correlates of diverse sorts of arousal, interpretations are limited to certain aspects (e. g. cognitive load and cognitive conflict) whilst others are hard to detect with such methods.

Furthermore, neurophysiological responses measured are indeed a result of diverse stimuli affecting the human body at the same time. That is why also isolating cause and effect relations as well as tracing back effects to root causes related to specific aspects of artifact design are challenging.

That said, interdisciplinary research projects appear suited to make best use of neuroscientific methods in future research. In such consortia, IS researchers can then pose questions for which appropriate theories are identified or experimental designs are developed together with neuroscientists. On the basis of such cooperations, a stronger methodological discussion on NeuroIS may take place in the medium term. This, however, requires the development of generally accepted quality criteria and procedures that can be used both in the work of authors and reviewers. As in other areas, the mere application of neuroscientific measurement techniques will certainly not be enough. Instead, IS researchers will have to learn using the new possibilities in a way that enables them to develop new knowledge surrounding the design and use of IT artifacts. Then, the opportunities are truly remarkable as we open up the scene to an entirely new source of data.

6 Conclusion

In this paper, we discussed the potentials of neuroscience for design-oriented IS research. We indicated areas of application for both the design and evaluation of IT artifacts. Not only can IS researchers use neuroscientific methods and tools in order to measure physiometric responses of people engaging in the usage or design of IT artifacts. There is also a wide range of neuroscientific theories that can be used in order to inform our research. We recommend complementing rather than substituting traditional research approaches. We, therefore, presented an example on triangulating with neuroscientific data in design-oriented research. Against this background we also identified weaknesses and limitations of neuroscience. With this we hope to contribute to a better understanding of how to make use of neuroscience in future design-oriented research.

References

1. Agarwal, R., Karahanna, E.: Time Flies When You're Having Fun: Cognitive Absorption and Beliefs about Information Technology. MIS Quarterly 24(4), 665–694 (2000)
2. Ambler, T., Braeutigam, S., Stins, J., Rose, S.P., Swithenby, S.J.: Salience and Choice: Neural Correlates of Shopping Decisions. Psychology & Marketing 21(4), 247–266 (2004)
3. Bandara, W., Gable, G.G., Rosemann, M.: Factors and Measures of Business Process Modelling: Model Building through a Multiple Case Study. European Journal of Information Systems 14(4), 347–360 (2005)
4. Batra, D., Marakas, G.M.: Conceptual Data Modeling in Theory and Practice. European Journal of Information Systems 4(3), 185–193 (1995)
5. Botvinick, M.M., Cohen, J.D., Carter, C.S.: Conflict Monitoring and Anterior Cingulate Cortex: An Update. Trends in Cognitive Sciences 8(12), 539–546 (2004)

6. Cacioppo, J.T., Tassinary, L.G., Fridlund, A.J.: The Skeletomotor System. In: Cacioppo, J.T., Tassinary, L.G. (eds.) Principles of Psychophysiology, pp. 325–384. Cambridge University Press, Cambridge (1990)
7. Camerer, C., Loewenstein, G.F., Prelec, D.: Neuroeconomics: How Neuroscience can Inform Economics. Journal of Economic Literature 43(1), 9–64 (2005)
8. Caroll, J.M.: Making Use: Scenario-Based Design of Human-Computer Interactions. MIT Press, Cambridge (2000)
9. Csikszentmihalyi, M.: Flow: The Psychology of Optimal Experience. Harper & Row, New York (1990)
10. Cuena, J., Molina, M.: The Role of Knowledge Modeling Techniques in Software Development: A General Approach Based on a Knowledge Management Tool. International Journal of Human and Computer Studies 52(3), 385–421 (2000)
11. Damasio, A.: Descarte's Error. Grosset/Putnam, New York (1994)
12. Davis, F., Banker, R.: The Technology Acceptance Model and Cognitive Neuroscience. In: Loos, P., Riedl, R., Müller-Pütz, G.R., vom Brocke, J., Davis, F.D., Banker, R.D., Léger, P.-M. (eds.) NeuroIS: Neuroscientific Approaches in the Investigation and Development of Information Systems (2010); Business & Information Systems Engineering 2(6), 395–491 (2010)
13. Dietrich, A.: The cognitive neuroscience of creativity. Psychonomic Bulletin & Review 11(6), 1011–1026 (2004)
14. Dimoka, A., Davis, F.D.: Where Does TAM Reside in the Brain? The Neural Mechanisms Underlying Technology Adoption. In: Proceedings of the International Conference on Information Systems, pp. 1–18 (2008)
15. Dimoka, A., Pavlou, P.A., Davis, F.D.: NeuroIS: The Potential of Cognitive Neuroscience for Information Systems Research. In: Information Systems Research (2010) (forthcoming) (published online in Articles in Advance, April 14, 2010)
16. Dimoka, A., Banker, R.D., Benbasat, I., Davis, F.D., Dennis, A.R., Dimoka, A., Gefen, D., Gupta, A., Ischebeck, A., Kenning, P., Müller-Putz, G., Pavlou, P.A., Riedl, R., vom Brocke, J., Weber, B.: On the Use of Neurophysiological Tools in IS Research: Developing a Research Agenda for NeuroIS. MIS Quarterly forthcoming (2011)
17. Fox, M.S., Gruninger, M.: Enterprise Modeling. AI Magazine 19(3), 109–121 (1998)
18. Freeman, F.G., Prinzel, L.J., Mikulka, P.J., Scerbo, M.W.: Evaluation of an Adaptive Automation System Using Three EEG Indices with a Visual Tracking Task. Biological Psychology 50(1), 61–76 (1999)
19. Freeman, F.G., Mikulka, P.J., Scerbo, M.W., Scott, L.: An Evaluation of an Adaptive Automation System Using a Cognitive Vigilance Task. Biological Psychology 67(3), 283–297 (2004)
20. Gregor, S.: The Nature of Theory in Information Systems. MIS Quarterly 30(3), 611–642 (2006)
21. Gregor, S., Jones, D.: The Anatomy of a Design Theory. Journal of the AIS 8(5), 312–335 (2007)
22. Haier, R.J., Siegel Jr., B.V., MacLachlan, A., Soderling, E., Lottenberg, S., Buchsbaum, M.S.: Regional Glucose Metabolic Changes after Learning a Complex Visuospatial/Motor Task: A Positron Emission Tomographic Study. Brain Research 570(1-2), 134–143 (1992)
23. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28(1), 75–105 (2004)
24. Kenning, P., Plassmann, H.: NeuroEconomics: An overview from an economic perspective. Brain Research Bulletin 67(5), 343–354 (2005)

25. Lee, N., Chamberlain, L.: Neuroimaging and Psychophysiological Measurement in Organizational Research: An Agenda for Research in Organizational Cognitive Neuroscience. Annals of the New York Academy of Sciences 1118, 18–42 (2007)
26. Léger, P.-M.: Using a Simulation Game Approach to Teach Enterprise Resource Planning Concepts. Journal of Information Systems Education 17, 441–448 (2006)
27. Léger, P.-M., Robert, J., Babin, G., Pellerin, R., Wagner, B.: ERPsim, ERPsim Lab (erpsim. hec.ca), HEC Montreal, QC. (2007)
28. Léger, P.-M., Charland, P., Robert, J., Babin, G.: Using a Business Simulation Game to Develop IT Competency. In: Temasek Polytechnic International Conference on Learning and Teaching (2010)
29. Loos, P., Riedl, R., Müller-Pütz, G.R., vom Brocke, J., Davis, F.D., Banker, R.D., Léger, P.-M.: NeruoIS: Neuroscientific Approaches in the Investigation and Development of Information Systems. Business & Information Systems Engineering 2(6), 395–491 (2010)
30. March, S.T., Smith, G.F.: Design and Natural Science Research on Information Technology. Decision Support Systems 15(4), 251–266 (1995)
31. Mendling, J., Reijers, H.A., Cardoso, J.: What Makes Process Models Understandable? In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 48–63. Springer, Heidelberg (2007)
32. Mineau, G.W., Missaoui, R., Godinx, R.: Conceptual Modeling for Data and Knowledge Management. Data & Knowledge Engineering 33(2), 137–168 (2000)
33. Nielsen, J.: Usability Engineering. Academic Press, Boston (1993)
34. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems 24(3), 45–77 (2008)
35. Pope, A., Bogart, E., Bartolome, D.: Biocybernetic System Evaluates Indices of Operator Engagement in Automated Task. Biological Psychology 40(1-2), 187–195 (1995)
36. Recker, J.: A Socio-Pragmatic Constructionist Framework for Understanding Quality in Process Modelling. Australasian Journal of Information Systems 14(2), 43–63 (2007)
37. Riedl, R.: Zum Erkenntnispotenzial der kognitiven Neurowissenschaften für die Wirtschaftsinformatik: Überlegungen anhand exemplarischer Anwendungen. NeuroPsychoEconomics 4, 32–44 (2009)
38. Riedl, R., Banker, R.D., Benbasat, I., Davis, F.D., Dennis, A.R., Dimoka, A., Gefen, D., Gupta, A., Ischebeck, A., Kenning, P., Müller-Putz, G., Pavlou, P.A., Straub, D.W., vom Brocke, J., Weber, B.: On the Foundations of NeuroIS: Reflections on the Gmunden Retreat 2009. Communications of the AIS 27, 243–264 (2010)
39. Riedl, R., Hubert, M., Kenning, P.: Are There Neural Gender Differences in Online Trust? An fMRI Study on the Perceived Trustworthiness of eBay Offers. MIS Quarterly 34(2), 397–428 (2010)
40. Riedl, R., Randolph, A.B., vom Brocke, J., Léger, P.-M., Dimoka, A.: The Potential of Neuroscience for Human-Computer Interaction Research. In: Proceedings of the 9th Annual Workshop on HCI Research in MIS, pp. 1–5 (2010)
41. Rossi, M., Sein, M.: Design Research Workshop: A Proactive Research Approach. Presentation delivered at IRIS 26, August 9-12 (2003)
42. Shaw, M.L.G., Gaines, B.R.: Requirements acquisition. Software Engineering Journal 11(3), 149–165 (1996)
43. Shoval, P., Shiran, S.: Entity-Relationship and Object-oriented Data Modeling: An Experimental Comparison of Design Quality. Data & Knowledge Engineering 21(3), 297–315 (1997)

44. Tan, D.S., Lee, J.C.: Using Electroencephalograph Signals for Task Classification and Activity Recognition. Microsoft Corporation, US Patent & Trademark Office, Serial No.: 349859, United States Patent Application No.: 20070185697 (2006), http://www.freepatentsonline.com/7580742.pdf
45. Tellegen, A., Atkinson, G.: Openness to Absorbing and Self-altering Experiences ("absorption"), a Trait Related to Hypnotic Susceptibility. Journal of Abnormal Psychology 83(3), 268–277 (1974)
46. Tuunanen, T., Peffers, K., Gengler, C.: Wide Audience Requirements Engineering (WARE). A Practical Method and Case Study. Sprouts: Working Papers on Information Systems 4(27) (2004), http://sprouts.aisnet.org/4-27
47. Vernadat, F.B.: Enterprise Modeling and Integration: Principles and Applications. Chapman & Hall, London (1996)
48. vom Brocke, J., Recker, J., Mendling, J.: Value-oriented Process Modeling: Integrating Financial Perspectives into Business Process Re-design. Business Process Management Journal 16(2), 333–356 (2010)
49. Walls, J.G., Widmeyer, G.R., El Sawy, O.: Assessing Information System Design Theory in Perspective: How Useful Was Our 1992 Initial Rendition? Journal of Information Technology Theory and Application 6(2), 43–58 (2004)
50. Webster, J., Ho, H.: Audience Engagement in Multi-media Presentations. The Database for the Advances in Information Systems 28, 63–77 (1997)
51. Winter, R.: Design Science Research in Europe. European Journal of Information Systems 17, 470–475 (2008)

On a NeuroIS Design Science Model

Christopher Liapis and Samir Chatterjee

Claremont Graduate University, School of Information Systems and Technology,
130 E 9th St., Claremont, CA 91711
chris@humanthreading.com, samir.chatterjee@cgu.edu

Abstract. In this paper, we present a novel frontier for IS research that we have termed "NeuroIS Design Science". Our study introduces a novel framework to the IS community which leverages neuroscience to better understand the design of human-computer interfaces. As a contribution to knowledge, the NeuroIS Design Science Model (NDSM) hopes to provide the scientific community with physiological measurements and thereby potentially advancing artifact design. This may serve as useful data to engineers, psychologists, neuroscientists, and manufacturers. What's more, the design and development of artifact creation could have a host of contributions in computer science, electrical engineering, as well as material sciences. With regard to information systems, this research presents a framework in human and interface interaction which does not currently exist. It allows researchers to follow a structure which may produce efficient technological artifacts for our future.

Keywords: Design Science, Neuroscience, Design Science Research Methodology, Human Threading, EEG, Information Systems.

1 Introduction

In this paper, we present a novel frontier for IS research that we have termed "NeuroIS Design Science". Within the discipline of IS, design science research has enabled the development of novel IT artifacts, organizational development, and theory building [25]. As a research methodology, the DSRM provides structure and direction to creative and often times loose concepts and artifacts [45]. Through these principles, design science has provided IS with advancement in socio- technical theory [21], knowledge processes [37], and an array of other contributions [28]. Furthermore, the plasticity of this domain allows other areas of research to naturally bind with core principles of IS. In effect, this study leverages such plasticity of open architecture in design science with neuroscience and human physiological values.

Neuroscience is the study of the nervous system [10]. The nervous system is an entire organ network which sends and receives signals triggering the action in all animals. The human has a central nervous system as well as a peripheral nervous system. The central nervous system for our discussion can be viewed as the brain and spinal cord. It is the human's information superhighway for communication. A neuron is a cell which is responsible for sending and receiving such information through electrical and chemical signals. The three main parts to a neuron are the soma,

H. Jain, A.P. Sinha, and P. Vitharana (Eds.): DESRIST 2011, LNCS 6629, pp. 440–451, 2011.

dendrite, and axon. At a high level, neural communication occurs when the nucleus instructs its axon to send a signal (neurotransmitter) down its nerve fiber. This signal is then received by other neurons whose dendrites are attached to the axon. The human brain has anywhere between fifty and one hundred billion neurons. They all connect to each other and generate multiple pathways which are termed neural networks. As a result of our neurobiology, science has created ways to measure these electrical and chemical signals.

These measurement techniques associated with our human physiology came to be through clinical research in health care. However, in the course of two decades researchers have begun using brain measuring techniques in non- clinical settings [3]. In many cases this was in an effort to better understand healthy human's cognitive, emotional, physical or behavioral characteristics. For example, stress and cognitive load have been observed among healthy men and women through electroencephalography [51].

The rest of the paper is organized as follows. In Section 2, we present a novel NeuroIS Design Science Model. In Section 3 we cover the relevant background literature that will help understand development of the artifacts and measuring physiological values. To demonstrate the usefulness of the model, we present in Section 4 the instantiation of a multi-modal interface artifact, the measurement experiments and results. Finally we conclude in Section 5 with thoughts on how this new frontier can contribute to IS research.

2 A NeuroIS Design Science Model

Building on previous work employing neuroscience and design science in IS [13] a NeuroIS Design Science Model is presented. The model proposes an architecture which utilizes neuroscience data in an effort to provide utility to design science researchers. There are three main regions to our presented model: Environment, NDSM, and the IS Knowledge Base (see Fig. 1).

In the Environment region of our model, a firm understanding in both principles of IS and fundamentals in neuroscience are required. This area is highly transdiciplinary as it forges neurophysiologic measurement techniques with computer interfaces. In this region an experiment is designed with an IS artifact set as its dependent variable. Once a thorough understanding of how an IT artifact is used in its native environment, researchers may design an experiment to test the efficacy of that artifact.

In the NDSM region, a three step IS design methodology is woven together recursively and based on neurophysiologic data. In the first step (based on the Environment region) a human is measured while using the IT interface as its variable. Next, our second step uses the measurement data as fortitude for creating a new IS interface or artifact. In this step, analysis of how the human reacted to the IT artifact will provide necessary data for which to build a more efficient version of the IT artifact. Efficiency may be measured through neurophysiologic values (cognitive load, stress, GSR, temperature, working memory, etc.), time complexities, and subjective assessments. Finally, another neurophysiologic test is conducted on a human; however this last step tests the IT interface or artifact that was developed in step two of this stage. The NDSM stage uses recursion between all three of its steps in effort to drive efficiency

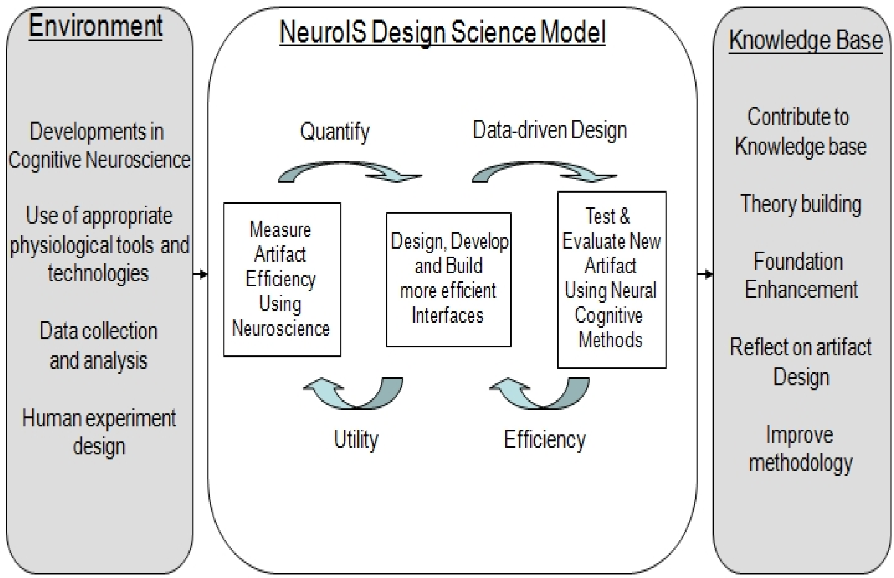


Fig. 1. A NeuroIS Design Science Model which portrays all three stages of measuring, designing, and testing a new IT interface or artifact

and utility. The primary focus of this stage is to measure, design, and then test a new IT interface or artifact.

In the Knowledge Base stage of this model, literature and information is disseminated back to the IS research community. As our NDSM is heavily transdiciplinary, there will be several key areas of research which need to be communicated back. For example, data on cohesion among participant's physiologic response, unique processes used to create a better interface or artifact, and the measurement findings on this new artifact.

Our model may be used across the IS spectrum for any number of phenomena under investigation. This may address areas of cognitive neuroscience focusing on stress, cognitive load, working memory, trust, cognitive enchantment, etc. These phenomena may then be tested in different environments inside of the IS domain. The primary goal behind this model is to understand areas of the body and brain in an effort to develop more neurologically efficient IT artifacts [13].

The printing area is 122 mm × 193 mm. The text should be justified to occupy the full line width, so that the right margin is not ragged, with words hyphenated as appropriate. Please fill pages so that the length of the text is no less than 180 mm, if possible.

3 Relevant Background Literature

Design science theory as a rigorous research guideline [26] and practical methodology [45] has been leveraged as the underling theory behind our design model. Design science (DS) was introduced into information systems and technology as a fundamental value based on stratified concepts of theory building and science of the artificial [18] [10]. It is understood in theory as a process which satisfies real world problems with logical and creative answers [30]. The applicable DS methodology in our model will be carried out using the design science research methodology (DSRM) [45]. This particular vehicle most efficiently fits the NDSM as it encompasses previous design science work on theory, [30] [31] applied engineering, [4] [29] and identification [36] with an emphasis on the Hevner and Chatterjee guidelines [25]. The DSRM was developed with six activities which use recursion to bridge from one to another. The power behind such a model is the inherent flexibility with regard to each stage in the framework.

Depending on the phenomena in a given NeuroIS experiment, an entry point for design could fall into any step and follow the process iteration. These activities include: problem identification, defining objectives, design and development, demonstration of the artifact, evaluation, and communication.

The first step in our model focuses on calculating and analyzing neural stimuli through physiologic measurement techniques. The problem identification will often times be novel and undiscovered. Primary motivation and justification behind this step may concentrate on an effort to reduce time completion or discovery of high cognitive load.

Cognitive load may be understood as the mental resources an individual exercises while solving problems or completing a task [49]. There has been a tremendous amount of research devoted to accurately deriving cognitive load from humans during task completion [1] [40]. Cognitive load theory [38] gave rise to investigation in educational instruction and psychological fatigue of the human brain during the 1970s. Later, this theory produced meaningful (and practical) discovery with the delineation of working memory, and cognitive overload [12]. Cognitive overload may be understood as the proliferation of data in short term memory which at a certain point overwhelms the individual.

As brain imaging and measurement technologies have flourished in recent years, so has the identification of key factors pointing to working memory and cognitive load. Electroencephalography (EEG) is an excellent example of how modern technology has accurate and fast indicators through post-synaptic potential measurement [9] [15]. There are both competing and complimentary brain-measuring techniques that researchers commonly use today from fMRI and EEG to infrared tomography and more [2] [14] [52]. One commonly used and accepted measurement of cognitive load is discovering event related potentials (ERP's) via electroencephalography [50]. ERP's are brain phenomena which produce multi-dimensional amplitudes (both positive and negative polarity) through time locked control of a given stimulus [35] [53]. Researchers often times introduce participants to a stimulus and look for unique potentials milliseconds after the stimulus has been presented. The NeuroIS Design Science Model will make use of the P300 and other ERP phenomena in future studies by gaining a baseline of a participant then introducing them to a given stimulus (IT interface in this case). The resultant stimulus will be analyzed milliseconds after the introduction [5] depending on phenomena.

Aside from P300, advancements in other neurological research areas have provided an array of measurement processes for studying the brain. Each specific measurement mechanism or process has its own particular set of advantages while simultaneously may offer some disadvantages [30]. Some of today's more popular methods include Magneto-encephalography, Single Photon Emission Computerized Tomography, Positron Emission Tomography, Functional Magnetic Resonance Imaging, and Functional Near-Infrared Imaging. As discussed, a common brain measurement methodology utilized in our model studies is EEG. This measurement system provides researchers with tremendous temporal resolution [3] [23] [19] [20] [22] [42], and is very important tool to the future of NeuroIS knowledge [16] [47] [17].

EEG research is a non-invasive procedure whereby electrodes placed on top of the scalp read neural activity (post-synaptic action potentials) on the outer most portion of the brain (neocortex) [48]. Post-synaptic action potential assessment via EEG has made significant contributions to Human Computer Interfaces and Brain Computer Interfacing as an accurate method in studying specific brain reaction to stimulus [50]. More specifically, EEG has proven itself as a reliable process in observing potentials through several modes of spectral phenomena beyond ERP's. For example measuring participant's pre-frontal theta activity alongside parietal alpha potentials has been proven a recurring gauge of cognitive load [3] [22] [48].

However, to its detriment, many studies reveal there is no silver bullet method to EEG and other brain measurement discovery [23] [27]. Brain measurement is still an imperfect science which offers what some researchers consider crude representations of neuronal activity [19]. Additional measurement techniques which may offset brain scans include galvanic skin response measurements, [39] [43] subjective workload questionnaires such as the NASA TLX [6] [44], electrocardiogram [11], saccade tests [33] and more.

4 A Multimodal Artifact Instantiation

To show as an example, how our proposed NDSM model is useful, we present a research project that investigated the effects email had on humans in relation to their workplace. Previous research estimates email to save companies 326 hours per employee every year ($9000 net savings per employee) over the use of telephone [45]. However, studies focusing on media richness theory counter this with evidence illustrating how communication decomposes as it deviates from voice [46][47]. That is to say it is more natural (and better understood by the recipient) to speak a message rather than typing it. As a result, our research team created a novel vestibular based system engineered to target speech and audition. We call our software "Aliis" which has taken years of development and we do not go into further details. This voice-based email system was invented to isolate speech and mitigate simultaneous reliance on additional modalities such as sight and touch (ergo keyboard, mouse, and LCD display). Thus, the scope of this study compared a standard Microsoft Outlook-based email system using Microsoft XP with a voice-based email system to determine two criteria. First, was it faster for participants to send an email using the standard PC/Outlook system they currently used at their workplace rather than the voice based system? And next, were there any differences in cognitive load between the systems? We obtained Institutional Review Board approval for conducting the measurement experiments.

To progress through our design model in a nominal sequence, time and cognitive load results produced a problem centered initiation at Step 1 in the NDSM. That is to say, researchers found current email systems time complexities unacceptable and hypothesized to reject its cognitive load values at the prefrontal and parietal cortices (Fz and Pz in the international 10–20 system). The investigation for a solution (and the objectives thereof) was established to lower cognitive load while speeding up the time to send an email (Step 2 in our model). This improvement could save businesses tremendous amounts of capital if hypothesis H0 were rejected (standard email system). A voice based artifact was developed in light of these prior two steps and demonstrated with 10 participants to meet the objectives set in Step 2. Finally, after a successful evaluation via EEG and time trial (Step 3) this data has been submitted as communication literature Knowledge Base Stage of our model.

4.1 Participants

All applicants were required to have a working knowledge of Microsoft Windows and Outlook 2007 through business practice. Informed consent was agreed to by all participants and each EEG session lasted approximately one and a half hours. Prior to each experiment participants were given a template consisting of the email they were to send (To, Subject, and Message data). This template required users to send the email to a generic Hotmail account with a subject of 'Antony'. The message data was two sentences used from The Tragedy of Julius Caesar by William Shakespeare. A brief walkthrough of each email task ensued in an effort to make sure participants understood their responsibilities.

4.2 EEG Measurements

EEG was recorded at all channels simultaneously using a Brainmaster Atlantis 4x4 with gold plated electrodes. Based on previous work, a monopolar montage was applied computing spectrograms on the International 10–20 system. Derivations of Fz and Pz were placed with a corresponding reference on A1 and A2 respectively [23] [7] [22] [3] [48]. Ground was set at Cz with a resistance rate of less than 5 kΩ. Our EEG sampling rate was set to 256 sps using Brainmaster 3.4 software with a peak to peak amplitude scale. Spectral power values set for theta (4–8 Hz) and alpha (8–12 Hz) was evaluated using a Fast Fourier transformation and all data was analyzed by means of 1 s epochs. Artifact was corrected for ocular and muscular interference through a built-in application analysis utility as well as manual inspection.

4.3 Results

The experiment was examined by comparing matched sets of an Outlook email task against a voice based email task. Each user conducted three sessions of one email task against another for a total of 30 matched email sets (n=10).Tasks were randomized between individuals in an effort to offset any unforeseen expectation or correlation. A matched set hypothesis t- test for the paired samples were assessed under the following assumption: $H_0{:}\mu_d \leq 0$ while H_1: $\mu_d > 0$. The t- test was carried out as follows $t = \bar{d} - \mu_d / s_{\bar{d}}$ where $s_{\bar{d}} = s / \sqrt{N}$ Both frontal theta and parietal alpha were investigated with degrees of freedom = 9 and confidence limits set to 95%

(critical value for α=.05) where the t- distribution= 2.262. With a calculated frontal theta value of 3.9966, the decision to reject H_0 is statistically significant as it surpasses the critical value 2.262. This result shows a tremendous difference in the activity of the theta wave inside our prefrontal cortex as the Outlook task produces a good deal more. However, the parietal cortex's alpha frequency does not reach the critical value for rejection (t = -2.6265).

With a sample size of 10 and sixty tasks (thirty competing between Outlook and voice), we can say with 95 percent confidence that the average frontal theta frequency modulates between 19.351 and 23.427 while using Outlook. However with the same confidence factor, speaking the email through the voice based system averages 19.648 to 16.052. This is statistically significant between the two systems showing the voice email as using a lower theta wave. Similarly, we can say with 95 percent confidence that the average range of our parietal alpha frequency lies between 8.538 and 6.868 while using Outlook. Speaking the email raised parietal alpha spectrum range averages from 9.94 to 7.634 (see Fig. 2).

The next test conducted measured time taken to complete a given email task (see Fig. 3). A two tailed t-test of equal variance was conducted to measure time with s_p^2 representing a pooled estimate. With degrees of freedom = 18, H_0:$\mu_1 - \mu_2 = 0$ and H1:$\mu_1 - \mu_2 \neq 0$. Confidence limits were set to 47.5 on each tail (α=.05) and the critical value was established at 2.10. Outlook tests were estimated in seconds as having a

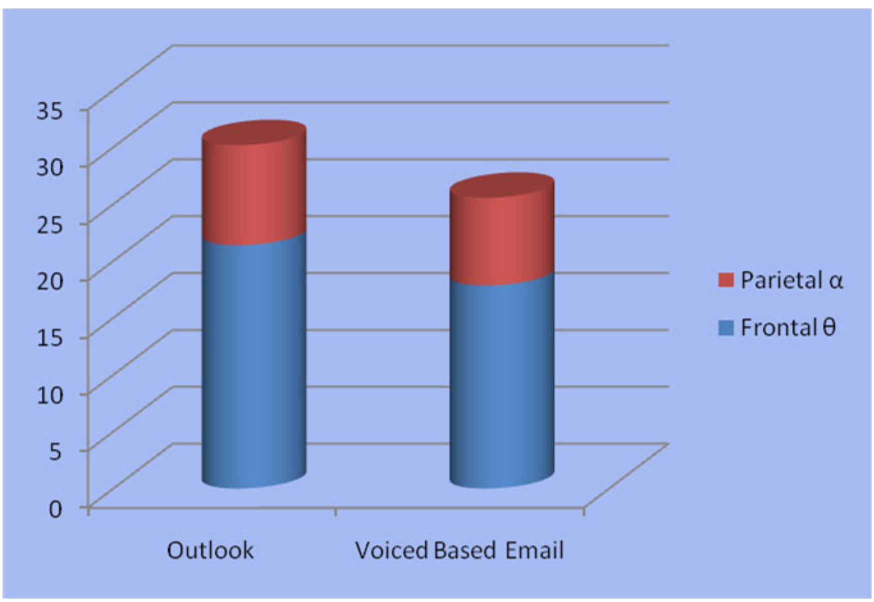


Fig. 2. Cognitive load measured in the prefrontal cortex with a focus on theta was significantly higher while using traditional type- based Outlook. Parietal alpha also increased, however this was not statistically significant.

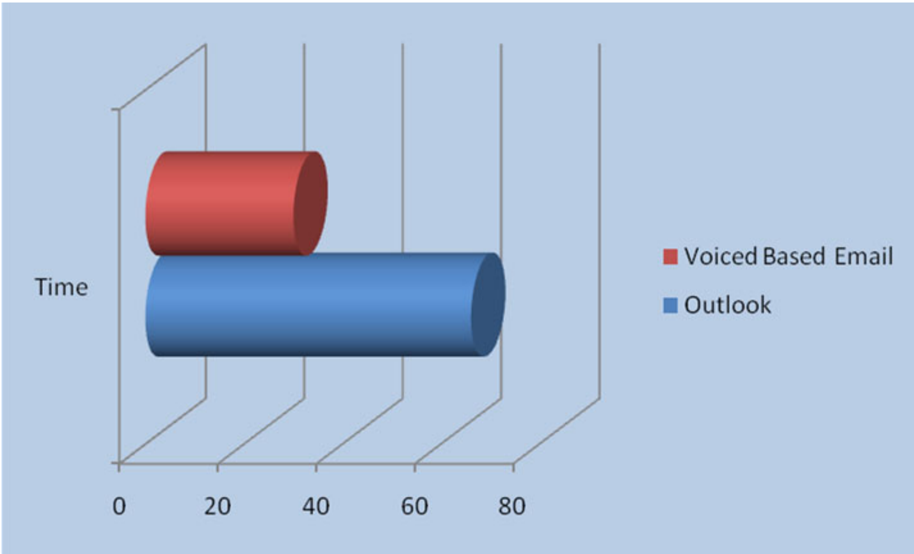


Fig. 3. Average time spent sending email. Outlook on average takes more than twice the time to send an email than the unfamiliar voice based system. Outlook Mean = 66.167 while voice based email Mean = 30.067.

mean of 66.1667 and standard deviation of 29.4619. Voice based mail had a mean of 30.0667 seconds and standard deviation of 4.697. As a result of the t-test, s_p^2 σ = 21.096 and resultant test statistic =3.83. As the calculated value of the test statistic is greater than our critical value, H_0 must be rejected. This data revels it was overwhelmingly faster to speak an email for the entire test population (10 out of 10) than type it.

5 Analysis and Findings

Our NeuroIS Design Science Model uses a process which measures human interaction with machines for future development of efficiency with systems. It aims to discover inefficient relationships and bridges these obstructions with artifacts. As a transdiciplinary applied system drawing on recursion and rooted in design theory this subject matter should generate new, practical technologies. The discovery offers researchers and designers an unprecedented window into human tendencies in relation to artifact usage. Conducting such research requires experts in many different areas of science and engineering to complete its mission accurately.

Furthermore, as human patterns of relational usage with machine are recorded using physiological analytics, our model hopes to discover methods and efficiencies currently unknown in IS research. The example presented in this paper is an exact case-in-point. Data has revealed several interesting findings which have direct implications on theory, development, and business practitioners. The original investigation sought to discern whether there was any utility gained by speaking an email rather than using our current approach of typing. Based on this study, our answer may quantifiably be yes.

Participants put forth much less mental effort while speaking an email (pre- frontal cortex) compared to using a standard personal computer and typing it. The prefrontal cortex almost unanimously showed a lower theta frequency with the voice based system compared to traditional typing in Outlook. However, with regard to the inverse effect of an increased parietal alpha, there was no statistical significance. That is, both frontal theta and parietal alpha seemed to decrease in frequency while speaking an email rather than typing it. What seemed most surprising about this result was participants had never before used the voice based system. Nonetheless, they seemed to reveal less mental effort in using it compared to a competing system they use at work every day. This result could mean many things from a neurological perspective. Perhaps the reason participants displayed smaller amplitude when speaking their email correlates to less mental effort using working memory and task switching [46].

The second metric used in measuring business utility also showed a statistical significance in favor of speaking an email. It was 220% faster on average for a participant to speak the email rather than type it. Again, this mean result was a consequence of participants' first interaction with a voice-based email system. Considering the favorable outcome on both dependant variables, a new question is posed: after using a voice-based email platform for a considerable amount of time, do participants spend even less time and cognitive load than revealed in this study? Nonetheless, the findings in this investigation point toward a greater utility earned by firms who adopt such a speech based email system. To be specific, currently companies who enjoy a saving of 326 hours a year (or approximately $9000 per employee) using a standard email platform could save 717 hours or about $20,000 through use of a voice based platform.

6 Conclusions

This study introduces a novel framework to the IS community which leverages neuroscience to better understand the design of human-computer interfaces. As a contribution to knowledge, the NeuroIS Design Science Model (NDSM) hopes to provide the scientific community with physiological measurements and thereby potentially advancing artifact design. This may serve as useful data to engineers, phsychologists, neuroscientists, and manufactureres. Whats more, the design and development of artifact creation could have a host of contributions in computer science, electrical engineering, as well as material sciences. With regard to information systems, this research presents a framework in human and interface or artifact interaction which does not currently exist. It allows researchers to follow a structure which may produce efficient technological artifacts for our future. NeuroIS does not pose the question of why humans design technologies. It is obvious every application created serves a purpose to assist society. However, whether that technology is efficient (with respect to congitive load, stress and enjoyment) is universally unanswered. Our work hopes to provide insights into new ways to develop artifacts for people, organizations, and society.

Another important aspect this research unlocks is a call to developers who produce software and hardware systems. Should this data be strengthened by future research with similar results, a voice-based email system may be highly sought after (from a financial viewpoint alone). Fabrication of such an artifact from the statistics presented in this study certainly may call for future development. However, this is not to impute such a system would have immediate characteristics for adoption. There are limitations to the development of this study and its technologies. For example, designers need to be cognizant of the immediate shortcomings associated with adopting a spoken language system. It may be disadvantageous to embrace such a system if employees work close to each other or if sound is a factor in the workplace. The voice-based email system used in this experiment would not be suitable in a real world production environment. It relies heavily on sparse kernel machines and only tolerates low background noise while sending an email. Another limitation which this experiment did not take into account is individual differences. The notion that future experiments encompass entire populations is highly unlikely. This research anticipates a contribution to knowledge from a gross prospective of participant data. Artifacts derived from our model should be made practical for many people, but certainly not all.

References

[1] Adey, W., Walter, D., Hendrix, C.: Computer techniques in correlation and spectral analyses of cerebral slow waves during discriminative behavior. Experimental Neurology 3, 501–524 (1961)

[2] Akgul, C.B., Akin, A., Sankur, B.: Extraction of cognitive activity-related waveforms from functional near-infrared spectroscopy signals. Medical & Biological Engineering & Computing 44(11), 945–958 (2006)

[3] Holm, A., et al.: Estimating Brain Load from the EEG. The Scientific World Journal 9, 639–651 (2009)

[4] Archer, B.: Design as a discipline. Design Studies 1(1), 17–20 (1979)

[5] Balconi, M., Lucchiari, C.: Consciousness and Emotional Facial Expression Recognition. Journal of Psychophysiology 21(2), 100–108 (2007)

[6] Botzer, A., Meyer, J.: The effect of effort on responses to binary cues. In: Proceedings of the 15th European Conference on Cognitive Ergonomics the Ergonomics of Cool Interaction - ECCE 2008, Presented at the 15th European Conference, Funchal, Portugal, p. 1 (2008)

[7] Brookings, J., Wilson, G., Swain, C.: Psychophysiological responses to changes in workload during simulated air traffic control. Biological Psychology 42(3), 361–377 (1996)

[8] Brunelle, E., Lapierre, J.: Testing media richness theory to explain consumers' intentions of buying online. In: Proceedings of the 10th International Conference on Electronic Commerce - ICEC 2008, Presented at the the 10th International Conference, Innsbruck, Austria, p. 1 (2008)

[9] Brunken, R., Plass, J., Leutner, D.: Direct Measurement of Cognitive Load in Multimedia Learning. Educational Psychologist 38(1), 53–61 (2003)

[10] Byrne, J., Roberts, J.: From Molecules to Networks, 2nd edn. An Introduction to Cellular and Molecular Neuroscience. Academic Press, Burlington (2009)

[11] Cao, Y., Theune, M., Nijholt, A.: Modality effects on cognitive load and performance in high-load information presentation. In: Proceedingsc of the 13th International Conference on Intelligent User Interfaces - IUI 2009, p. 335 (2009)

[12] Chandler, P., Sweller, J.: The split-attention effect as a factor in the design of instruction. British Journal of Educational Psychology 62, 233–246 (1992)

[13] Liapis, C.: A Primer to Human Threading. Computers in Human Behavior 27(1), 138–143 (2011)

[14] Cordes, D., et al.: Frequencies contributing to functional connectivity in the cerebral cortex in "resting-state" data. AJNR. American Journal of Neuroradiology 22(7), 1326–1333 (2001)
[15] Curtis, C.: Persistent activity in the prefrontal cortex during working memory. Trends in Cognitive Sciences 7(9), 415–423 (2003)
[16] Dimoka, A., Benbasat, I., Lim, K.H., Straub, D.W., Walden, E.: NeuroIS: Challenges and Solutions. In: Proceedings of the 31st International Conference on Information Systems, pp. 1–5 (2010)
[17] Dimoka, A., Pavlou, P.A., Davis, F.D.: NEURO-IS: The Potential of Cognitive Neuroscience for Information Systems Research. In: Proceedings of the 28th International Conference on Information Systems, pp. 1–20 (2007)
[18] Dubin, R.: Theory building. Free Press, New York (1969)
[19] Ferrez, P., Millán, J.: You Are Wrong!—Automatic Detection of Interaction Errors from Brain Waves. In: Proceedings of the 19th International Joint Conference on Artificial Intelligence, Edinburgh, UK (2005)
[20] Gerě, I., Jaušcvec, N.: Multimedia: Differences in cognitive processes observed with EEG. Educational Technology Research and Development 47(3), 5–14 (1999)
[21] Germonprez, M., Hovorka, D., Collopy, F.: A Theory of Tailorable Technology Design. Journal of the Association for Information Systems 8(6), article 21 (2007)
[22] Gevins, A., et al.: High-resolution EEG mapping of cortical activation related to working memory: effects of task difficulty, type of processing, and practice. Cerebral Cortex 7, 374–385 (1997)
[23] Gevins, A., et al.: Monitoring working memory load during computer based tasks with EEG pattern recognition methods. Human Factors: The Journal of the Human Factors and Ergonomics Society 40(1), 79–91 (1998)
[24] Simon, H.: The Sciences of the Artificial, 3rd edn. MIT Press, Cambridge
[25] Hevner, A., Chatterjee, S.: Design Research in Information Systems: Theory & Practice. Springer Publisher Inc., Heidelberg (May 2010)
[26] Hevner, A., March, S., Park, J., Ram, S.: Design Science Research in Information Systems. Management Information Systems Quarterly 28(1), 75–105 (2004)
[27] Hruby, T., Marsalek, P.: Event-related potentials–the P3 wave. Acta Neurobiol. Exp. (Wars) 63(1), 55–63 (2003)
[28] Indulska, M., Recker, J.C.: Design science in IS research: a literature analysis. In: 4th Biennial ANU Workshop on Information Systems Foundations, Canberra, Australia, October 2-3 (2008)
[29] Eekels, J.: A methodological comparison of the structures of scientific research and engineering design: their similarities and differences. Design Studies 12(4), 197–203 (1991)
[30] Nunamaker, J.F., Chen, M.: Systems development in information systems research. In: Twenty-Third Annual Hawaii International Conference on System Sciences, presented at the Twenty-Third Annual Hawaii International Conference on System Sciences, Kailua-Kona, HI, USA, pp. 631–640.
[31] Walls, J.G., Widmeyer, G.R., El Sawy, O.A.: Building an Information System Design Theory for Vigilant EIS. Information Systems Research 3(1), 36–59 (1992)
[32] Korzeniowski, P.: Value of E-Mail? About $9,000 Per Worker, Investor's Business Daily (May 25, 2000)
[33] Lee, C., Rohrer, W., Sparks, D.: Population coding of saccadic eye movements by neurons in the superior colliculus. Nature 332(6162), 357–360 (1988)
[34] Lee, J., Tan, D.: Using a low-cost electroencephalograph for task classification in HCI research. In: Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology - UIST 2006, Presented at the the 19th Annual ACM Symposium, Montreux, Switzerland, p. 81 (2006)

[35] Luck, S.: An introduction to the event-related potential technique. MIT Press, Cambridge (2005)
[36] Rossi, M., et al.: Design Research workshop: A proactive research approach. In: Twenty-Sixth Information Systems Research Seminar, presented at the Design Research Workshop IRIS26, Haikko Finland (2003)
[37] Markus, M.L., Majchrzak, A., Gasser, L.: A Design Theory for Systems that Support Emergent Knowledge Processes. MIS Quarterly 26(3), 179–212 (2002)
[38] Marsh, P.: The instructional message: A theoretical perspective. Educational Technology Research and Development 27(4), 303–318 (1979)
[39] McCleary, R.: The nature of the galvanic skin response. Psychological Bulletin 47(2), 97–117 (1950)
[40] Michels, L., et al.: Simultaneous EEG-fMRI during a Working Memory Task: Modulations in Low and High Frequency Bands. In: Herzog, M.H. (ed.) PLoS ONE, vol. 5(4), p. e10298 (2010)
[41] Media richness theory - Wikipedia, the free encyclopedia, `http://en.wikipedia.org/wiki/Media_richness_theory`
[42] Nunez, P., Srinivasan, R.: Electroencephalogram - Scholarpedia. Electroencephalogram, `http://www.scholarpedia.org/article/Electroencephalogram`
[43] Paintal, A.: A comparison of the galvanic skin responses of normals and psychotics. Journal of Experimental Psychology 41(6), 425–428 (1951)
[44] Park, S., Harada, A., Igarashi, H.: Influences of personal preference on product usability. In: CHI 2006 Extended Abstracts on Human Factors in Computing Systems - CHI 2006, Presented at the CHI 2006 Extended Abstracts, Montreal, Quebec, Canada, p. 87 (2006)
[45] Peffers, K., Chatterjee, S., Tuunanen, T., Rothenberger, M.: A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems 24(3), 45–77 (2007)
[46] Ramnani, N., Owen, A.: Anterior prefrontal cortex: insights into function from anatomy and neuroimaging. Nature Reviews Neuroscience 5(3), 184–194 (2004)
[47] Riedl, R., Banker, R.D., Benbasat, I., Davis, F.D., Dennis, A.R., Dimoka, A., Gefen, D., Gupta, A., Ischebeck, A., Kenning, P., Müller-Putz, G., Pavlou, P.A., Straub, D.W.: On the Foundations of NeuroIS: Reflections on the Gmunden Retreat 2009. Communications of the AIS 27, 243–264 (2010)
[48] Sauseng, P., et al.: Fronto-parietal EEG coherence in theta and upper alpha reflect central executive functions of working memory. International Journal of Psychophysiology: Official Journal of the International Organization of Psychophysiology 57(2), 97–103 (2005)
[49] Oviatt, S.: Human-centered design meets cognitive load theory. In: Proceedings of the 14th annual ACM International Conference on Multimedia - MULTIMEDIA 2006, Presented at the 14th Annual ACM International Conference, Santa Barbara, CA, USA, p. 871 (2006)
[50] Shenoy, P., Tan, D.: Human-aided computing. In: Proceeding of the Twenty-Sixth Annual CHI Conference on Human Factors in Computing Systems - CHI 2008, Presented at the Proceeding of the Twenty-Sixth Annual CHI Conference, Florence, Italy, p. 845 (2008)
[51] Stipacek, A., et al.: Sensitivity of human EEG alpha band desynchronization to different working memory components and increasing levels of memory load. Neuroscience Letters 353(3), 193–196 (2003)
[52] White, B., et al.: Resting-state functional connectivity in the human brain revealed with diffuse optical tomography. NeuroImage 47(1), 148–156 (2009)
[53] Zani, A., Proverbio, A.: The Cognitive Electrophysiology of Mind and Brain. Academic Press, San DIego (2003)

The Value of Anonymity on the Internet

Dongwon Lim, Hangjung Zo*, and Dukhee Lee

Management Science Department, KAIST
Guesong-dong, Yuseong-gu, Daejon, Republic of Korea
{idw111,joezo,dukheelee}@kaist.ac.kr

Abstract. As anonymity has both positive and negative effects at the same time, it is arguable whether or not anonymity is worth preserving. However, there are few studies that seek to clarify the effects of anonymity on the society as a whole by integrating individual behaviors and macroscopic models.We propose an opinion diffusion model that introduces a 'conviction' dimension to represent behaviors of an anonymous agent,and investigate the way anonymity can affect the society using simulation method. Results indicate that anonymity is more effective in a society with a higher similarity threshold. In addition, increasing anonymity resulted in increasing the time to reach consensus and increasing the number of agents in the biggest cluster.

Keywords: anonymity; opinion dynamics; agent-based model; privacy; Internet.

1 Introduction

There is a general consensus that anonymity must be preserved to guarantee freedom of expression and protect privacy. However, we have experienced many consequences from anonymity, especially on the Internet. For instance, the Korean pop star Daniel SeonWoong Lee was confronted with a campaign by a group of Internet usersattempting to discredit his academic achievements. There are more cases in which online anonymity is exploited toattack others.

Several studies suggest that anonymity can give rise to anti-normative and anti-social behaviors. Mann[16] found that, in a baiting situation in which a personis about to commit a suicide by jumping off a building, factors related to anonymity such as crowd size, cover of darkness, and physical distance between victim and crowd tempted the crowd to incite the victim to jump. In addition, anonymity could be one of the means by which"online fraud operators are able to strike quickly, victimize thousands of consumers in a short period, and disappear without trace" [2](p. 592). Davenport [10] also states that online anonymity facilitates Internet-based crimes, "such as hacking, virus writing, denial-of-service attacks, credit card fraud, harassment, and identity theft" (p. 34).

On the other hand, some studies shed light on the positive aspects of anonymity. Christopherson[8] suggested that anonymity is one the most important means of

* Corresponding author.

H. Jain, A.P. Sinha, and P. Vitharana (Eds.): DESRIST 2011, LNCS 6629, pp. 452–464, 2011.

protecting privacy for psychological wellbeing. She pointed out that, by hiding one's physical appearance, one can be free from discrimination "based on gender, race, age, ethnicity, physical disability, and attractiveness" (p. 3045). Additionally, in a typical whistle-blowing situation, where the power differentials between the whistleblower and the accused are enormous, anonymity could encourage people to come forward and speak up the truth, and that would eventually promote the public welfare[11]. The participants in an American Association for the Advancement of Science (AAAS) conference considered anonymous communication a strong human right, and generally agreed that it should be discretionary for individuals and organizations to determine their level of anonymity [20].

Many researchers agree that anonymity can benefit or damage the society. Suler[19]distinguished between benign and toxic disinhibition that comes from anonymity.Teich et al. [20] suggested that anonymity is like a double-edged sword, and thus "it is important to distinguish among uses and types of anonymous communication so that the evils of one form do not serve as reasons for unnecessarily restricting others" (p. 72). In other words, as anonymity has positive and negative effects at the same time, it is arguable whether or not anonymity is worth preserving. If academic and public discussion determines that anonymity must be preserved, the extent to which it should be allowed must also be discussed.

Researchers have been trying to identify the characteristics of anonymity. Some researchers focused on the effects of anonymity on human behavior [8, 14,15, 16,19], and others proposed the degree to which anonymity should be preserved or regulated [10, 20]. However, the empirical studies have soughtonly to reveal the motivation behind individual behaviors. Researchers who suggestedanonymity policies have borrowed their rationale mostly from only one side, and their suggestions are still conflicting. While some researchers proposed models explaining how opinions diffuse among numerous agents [1, 3, 4, 13, 21], there have been no efforts to explain how anonymity could affect the diffusion. In short, there is little research that seeks to clarify the effects of anonymity on the society as a whole by integrating individual behaviors into macroscopic phenomena.

This study proposes an opinion diffusion model that uses conviction as a new dimension for expressing opinion. By introducing conviction, we seek todistinguish behaviors of anonymous agents and real-name agents. Weaddress theories and empirical studies from psychology, cognitive science and behavioral science in order to support and justify the model. We thenrun a simulation to identify how anonymity affects the whole network, and suggest some implications from the analysis of the simulation results.

2 Opinion Diffusion Models

2.1 Axelrod's Model of Disseminating Culture

Axelrod's model [4] "describes a culture as a list of features or dimensions of culture. For each feature there is a set of traits, which are the alternative values the feature may have" (p. 208). In the model, a site with several cultural features is selected randomly, and one of its cultural features is transferred to one of its neighbor sitesselected based on the probability equal to their cultural similarity.Using a simulation method, Axelrod showed that the cultural sites are globally segregated while they tend to converge locally.

2.2 Bounded Confidence Model

Weisbuch et al. [21] introduced the bounded confidence model in which "agents adjusts continuous opinions as a result of random binary encounters whenever their difference in opinions is below a given threshold."In the bounded confidence model, each agent has its own opinion x and an uncertainty level u about its opinion. Neighboring agents whose opinion differs by u can influence the agent. Weisbuch et al. [21] showed that randomly dispersed opinions finally converge into a single cluster for a large uncertainty level (u> 0.3), and several clusters is observed for a lower uncertainty level.

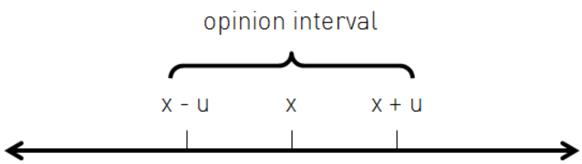


Fig. 1. An agent has its own opinion x, and uncertainty level u about its opinion. If neighboring agents have opinion within the opinion interval, they can influence the agent.

2.3 Vector Opinion Model

Weisbuch et al. [21] also suggested an opinion vector model, which treats opinions as binary. In the model, an agent has m binary opinions about the complete set of m subjects in a vectorform. This model is similar to Axelrod's model in characterizing agents with a set of integer vectors on a square lattice. "The adjustment process occurs when agents agree on at least $m–d$ subjects, where d denotes a discrepancy threshold. The rules for adjustment are as follows: when opinions on a subject differ, one agent is convinced by the other agent with probability μ" [21] (p. 62).

While these opinion dynamics models are successful to explainhow opinions are disseminated and consensus is reached, they lack the capability of demonstrating how anonymity can affect the process. Here we propose an opinion dynamics model that can explain anonymous agent's behavior.

3 Model Proposal

3.1 Dimensions of Opinion

Brucks[7] distinguished between objective and subjective knowledge. "Subjective knowledge can be thought of as including an individual's degree of confidence in his/her knowledge, while objective knowledge refers only to what an individual actually knows" (p. 2).

Previous models treated opinions as one dimension, which is objective knowledge or what an agent thinks or knows. In the proposed model, an opinion comprises two dimensions; information (i.e. objective knowledge or what an agent actually knows) and conviction (i.e. subjective knowledge or how confident the agent is about the information). Both dimensions are coded as 0 (unconvinced for conviction and false for information) or 1 (convinced for conviction and true for information). There are four types of opinions with two binary dimensions; convinced true, convinced false, unconvinced true, and unconvinced false.

	false (0)	true (1)
unconvinced (0)	UF 0	UT 1
convinced (1)	CF 2	CT 3

Fig. 2. An opinion comprises two dimensions; subjective knowledge and objective knowledge. Each dimension is coded as binary, so there are 4 types of opinions.

3.2 Rules

In order to investigate the effects of anonymity on opinion diffusion, the agent's state is divided into real-name and anonymous. Real-name and anonymous agents will behave differently according to their state. The rules mainly comprisethe different behaviors of the agents.

Rule 1. *A real-name agent will diffuse only convinced opinions, while an anonymous agent will diffuse both convinced and unconvinced opinions.*

The first rule issupported by a theory about social anxiety and self-presentation. Self-presentation is the attempt to show desirable impression of self to others, and if a person thinks that he will fail to do so, he feels social anxiety[18]. If an agent diffuses something wrong, he can get a bad reputation for it.Real-name agents will be reluctant to diffuse unconvinced opinions due to social anxietybecause they are easily identifiable. On the other hand, anonymous agents will not experience such an uncomfortable feeling and will thus be able to diffuse unconvinced opinions without reserve because they are not traceable.

The disinhibition effect also accounts for the behavior of anonymous agents. Disinhibition is a phenomenon in which "people self-disclose or act out more frequently or intensely than they would in person"[19]. Disinhibition is attributed to several elements, and dissociative anonymity is one of the elements. That is, dissociative anonymity could bring out unordinary self-disclosure which may include unconvinced opinions.

Rule 2. *A real-name agent has more credibility than an anonymous one, and therefore agents are more likely to believe information from a real-name source.*

When an anonymous agent deceives others, it's difficult to track him down. Therefore, people are less likely to believe an agent when he does not reveal himself to others. Corritore et al. [9] identified four dimensions of credibility: honesty, expertise, predictability and reputation.Honesty and expertise can be identified from the opinion itself. However, predictability and reputationcannot be measured for anonymous agents, and low credibility can also be attributed to them.

$$C_r > C_a \tag{1}$$

C_r Credibility of a real-name source (0.0 – 1.0)
C_a Credibility of an anonymous source (0.0 – 1.0)

Rule 3. *Convictionsof an agent can be changed by interactions with other agents. Convinced opinions rarely change to unconvinced, and unconvinced opinions easily change to convinced.*

We assume that there is a phase transition in the process of changing one belief to another. In Figure 3, only an unconvinced opinion can be changed to another opinion. Conviction could change, too. 'Convinced' opinions can be changed to 'unconvinced', and 'unconvinced' can be changed to 'convinced'. However, the likelihood of changing conviction is asymmetric due to cognitive dissonance and selective perception.

A person may take one of two stepsin order to reduce cognitive dissonance; conformity or disparagement. "Conformity is the preferred mode of dissonance reduction at the slight and moderate discrepancies" and "conformity drops out at the extreme discrepancies" [6] (p. 614). Therefore, once a conviction is established, it rarely changes. Moreover, in a psychological experiment where subjects were given an ambiguous figure and were required to interpret it,the subjects explained it according to their preferences[5].This means that people tend to believe what they want to believe. Therefore, 'unconvinced' opinions can easily be changed to 'convinced' opinions. In addition, source credibility can affect P_c and P_u.

$$P_c > P_u \tag{2}$$

P_c Probability of changing conviction from 'unconvinced' to 'convinced'
P_u Probability of changing conviction from 'convinced' to 'unconvinced'

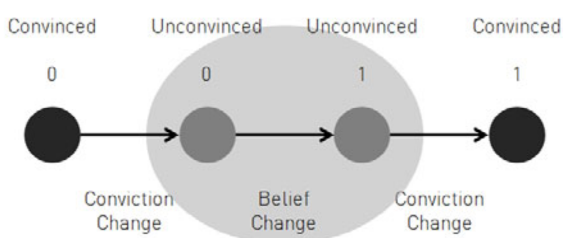


Fig. 3. In the process of belief transition, convinced opinions can be changed to unconvinced opinions, and unconvinced opinions can be changed to the opposite opinion.Also, unconvinced opinions can be changed to convinced opinions.

Rule 4. *Similarity between agents will affect the interaction between them.Agents will interact with only those who have similarity beyond their threshold.Similarity between anonymous users is perceived to be higher than it actually is.*

According to the theory of cognitive dissonance, an increasing discrepancy between communicating agents produces increasing dissonance. Generally, people have motivational drive to reduce dissonanceby any of four ways: conform to the communicator's point of view, disparagethe communicator, persuade the communicator that he is correct, and obtainsocial support from other like-minded individuals. "Theoretically, increasing discrepancy should result in increasing conformity at the moderate discrepancies, and increasing disparagement at the extreme discrepancies" [6].Conformity implies that a communicatee is likely to change his or her opinion to a communicator's, and disparagement implies that a communicatee will not care about the communication and keep his or her opinion as before. In borderline areas where increasing discrepancy could result in either conformity or disparagement, there lies a discrepancy threshold, or a similarity threshold.

In the proposed model, a discrepancy between two communicating agents can be obtained by calculating the proportion of the number of different opinions to the total number of total opinions. Similarly, we can adopt a concept of similarity, which captures how many opinions the two communicating agents have in common.

$$D_{ij} = \Sigma_k(x_{ik} - x_{jk}) / N \quad (3)$$

As similarity is the opposite of discrepancy, the equation can be rewritten as follows.

$$S_{ij} = 1 - D_{ij} \quad (4)$$

D_{ij} Discrepancy between *i*th agent and *j*th agent
S_{ij} Similarity between *i*th agent and *j*th agent
x_{ik} Opinion about *k*th topic of *i*th agent ($x_{ik} = 0$ or 1)
x_{jk} Opinion about *k*th topic of *j*th agent ($x_{jk} = 0$ or 1)
N Number of opinions an agent has

Rule 4 states that similarity between anonymous agents is perceived differently from similarity between real-name agents. Psychological theory can help us justify the rule. According to the social identity of deindividuation effect (SIDE) model, factors such as "the combination of anonymity and group immersion can actually reinforce group salience and conformity to group norms, and thereby strengthen the impact of a variety of social boundaries"[17] (p. 697).

Also, in an experiment of Lee[15], subjects were told information on other participants, such as major, age, and favorite films and music, before they took part in a discussion. As the SIDE model anticipated, subjects without information on the others showed stronger group identification. The result of the experiment "confirmed that deindividuated participants felt greater within-group similarity than did the individual participants"[15] (p. 393). In other words, the experiment suggests that, in an anonymous environment, people can feel more similar to each other than in an identifiable environment.

$$P_b^{\,r} = 0 \text{ (if } S_{ij} < T\text{), and } P_b^{\,r} = C_r \text{(if } S_{ij} \quad T\text{)} \tag{5}$$

$$P_b^{\,a} = 0 \text{ (if } S_{ij} < T\text{), and } P_b^{\,a} = C_a \text{(if } S_{ij} \quad T\text{)} \tag{6}$$

T Threshold of Similarity
$P_b^{\,a}$ Probability of changing belief in accordance with an anonymous agent
$P_b^{\,r}$ Probability of changing belief in accordance with a real-name agent

4 Simulation

In our simulations, a random scale-free network is builtat the beginning of a run. At each time step, an agent is selected based on the number of links it has. In other words, an agent with more links has a higher probability of being selected. Then the selected agent diffuses one of its opinions to all of its neighbors. If the similarity between the selected agent and each of its neighbors is larger than the similarity threshold, the neighbor changes its belief to conform to the incoming opinion stochastically. The simulation goes on until the system reaches an equilibrium state. An equilibrium state is defined as ten consecutive local equilibriums, where a local equilibrium refers to the state in which less than or equal to one belief change occurs during 1,000 time steps.

Table 1. Parameter settings for simulations

Parameter	Value	Remarks
N (number of nodes)	100	Scale-free network
K (number of links)	200	
Cr		
Ca		
Pc		Multiplied by C_r or C_a
Pu		
T	0.6, 0.8	
Anonymity proportion	0.0 – 1.0	0.2 interval
Number of opinions per agent	10	
Initial conviction	0.4, 0.6, 1.0	
Initial convinced true opinions	0.6, 0.8	
Initial unconvinced true opinions	0.6, 0.8	

5 Results

In Figure 4 and Figure 5, the X axis representsthe anonymity proportion. That is, at the zero anonymity level, there is no anonymous agent in the network at all, and at the full anonymity level only anonymous agents exist. At the intermediate proportions, real-name and anonymous agents exist together. The Y axis represents the number of ‘convinced’ and ‘true’ opinions. The number 1,000 is equal to 100% as one agent has ten opinions and one hundred agents were used in the simulation.

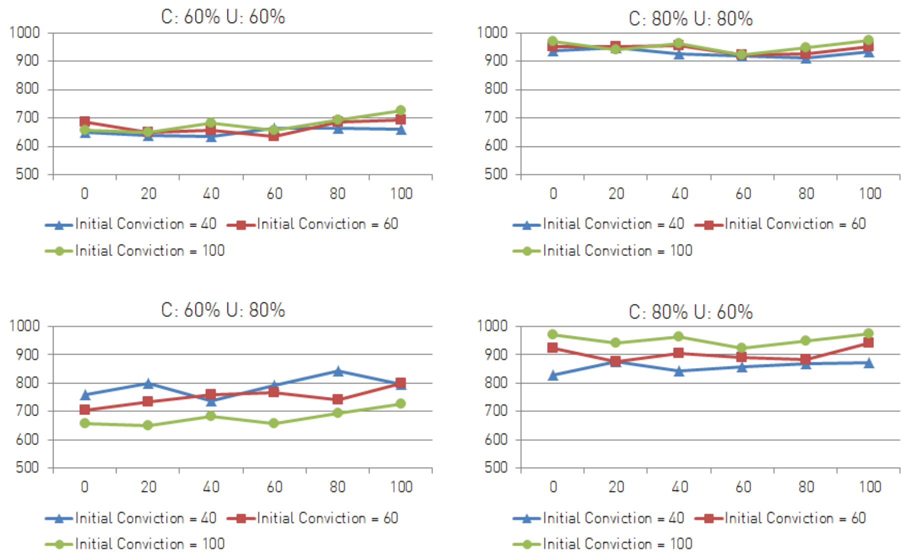


Fig. 4. In a network where similarity threshold is 0.6, graph lines are relatively flat, which means that there is little difference between no anonymity (0%) and full anonymity (100%)

In the upper side of each graph, there is a legend which shows how many convinced true opinions (C) and unconvinced true opinions (U) exist at the beginning. In the graph, each line stands for the different initial conviction, which describes the proportion of convinced opinions at the beginning.

Table 2. One-way ANOVA at similarity threshold 0.6 shows that the anonymity does not bring out significant difference in the number of convinced true opinions (initial conviction = 0.4, initial convinced true = 0.6, initial unconvinced true = 0.6)

Anonymity	N	mean	Std. dev.	dof	F	Sig
0	30	648.07	107.745			
20	30	639.90	123.534			
40	30	634.43	121.222	df1 = 5		
60	30	665.07	103.038	df2 = 174	.340	.888
80	30	664.33	135.622	total = 179		
100	30	660.37	144.352			
Total	180	652.03	122.288			

In Figure 4, where the similarity threshold is 0.6, an increase in anonymity level resulted in little difference. On the other hand, in Figure 5 where the similarity threshold is 0.8, an increase in the number of 'convinced true' opinions is salient with increasing anonymity.

Furthermore, statistical analysis shows that the difference in results is not significant at similarity threshold 0.6, while it is significant at similarity threshold 0.8. That is, anonymity has more impact in a higher threshold environment.

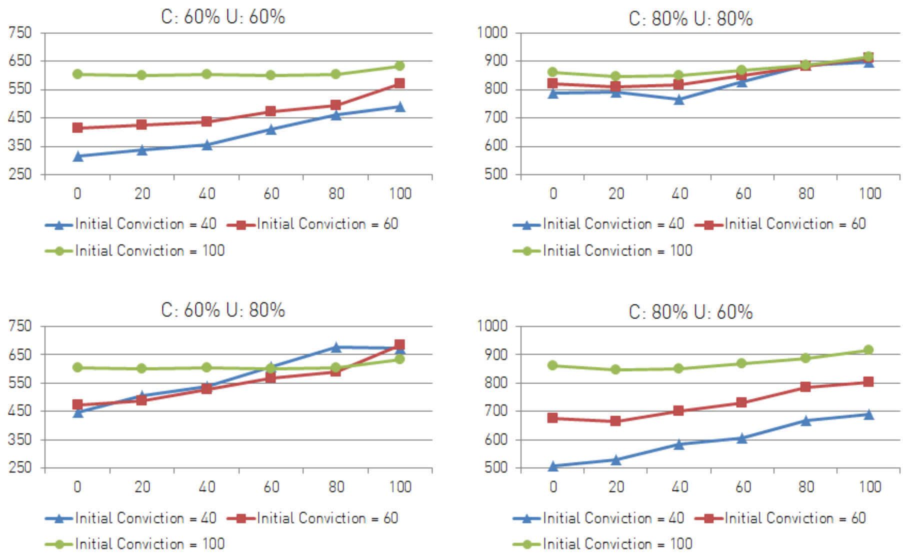


Fig. 5. In a network where similarity threshold is 0.8, graph lines are relatively flat, which means that there is little difference between no anonymity (0%) and full anonymity (100%)

Table 3. One-way ANOVA at similarity threshold 0.8 shows that the anonymity bring out significant difference in the number of convinced true opinions (initial conviction = 0.4, initial convinced true = 0.6, initial unconvinced true = 0.6)

Anonymity	N	mean	Std. dev.	dof	F	Sig
0	30	317.17	32.743			
20	30	337.07	37.486			
40	30	354.37	32.292	df1 = 5		
60	30	410.97	47.529	df2 = 174	81.967	.000
80	30	460.83	49.718	total = 179		
100	30	489.63	50.573			
Total	180	395.01	76.653			

Why are the effects of anonymity more salient in a society with a higher similarity threshold? In a society with a high similarity threshold, people will communicate with only those who have very similar opinions with themselves, and they may have few neighbors to communicate with. On the other hand, in a society with low similarity threshold, people are willing to talk to each other even though they are very different in their opinions.

By becoming anonymous, an agent can overstate its similarity, and thus able to communicate with others who otherwise would have not listened to its opinions. In a closed society where the similarity threshold is high, agents can rarely find those whom they can communicate with because other agents will turn their back even when there is a small difference. Therefore, the benefit of becoming anonymous is huge in a closed society. On the other hand, in an open society where the similarity threshold is low, a real-name agent can find many others who would like to converse with him. People will embrace others whose opinions are very different from theirs in such a society. Therefore, in an open society, becoming anonymous only makes an agent less credible. Even if anonymity enables an agent to connect more to other anonymous agents, the newly added connections are not necessarily better in terms of opinion quality. Even when their opinions are as good as other agents', low credibility of the anonymous agents will make the conviction process slower.

We should also focus on the time it takes to reach an equilibrium state. In general, increasing anonymity resulted in increasing the time to reach equilibrium. In all settings, the time to equilibrium increased drastically at the full anonymity level compared to zero anonymity. As agents are less likely to believe an anonymous agent, opinions are likely to be kept unchanged in an environment where there are many anonymous agents in the system. We can say that anonymity makes the system inefficient in terms of the time it takes to reach consensus.

Another interest of our research is the number of agents in the biggest cluster. The importance of this measure can be explained by the following example from Aldashev&Carletti[3]. Consider a community that is confronted by having to make a decision collectively by the majority rule on some issue. Initially, the citizens may have very different opinions about the issue. However, before the vote takes place, citizens will discuss the issue among themselves to reach consensus. If, at the end of the discussion, the biggest party includes only a few citizens, the outcome of the vote will leave most citizens disappointed. On the other hand, a community where the biggest party includes a large population will be capable of satisfying the majority of the citizens. Figure 7 shows that increasing the anonymity in the system also increased the number of agents included in the biggest cluster.

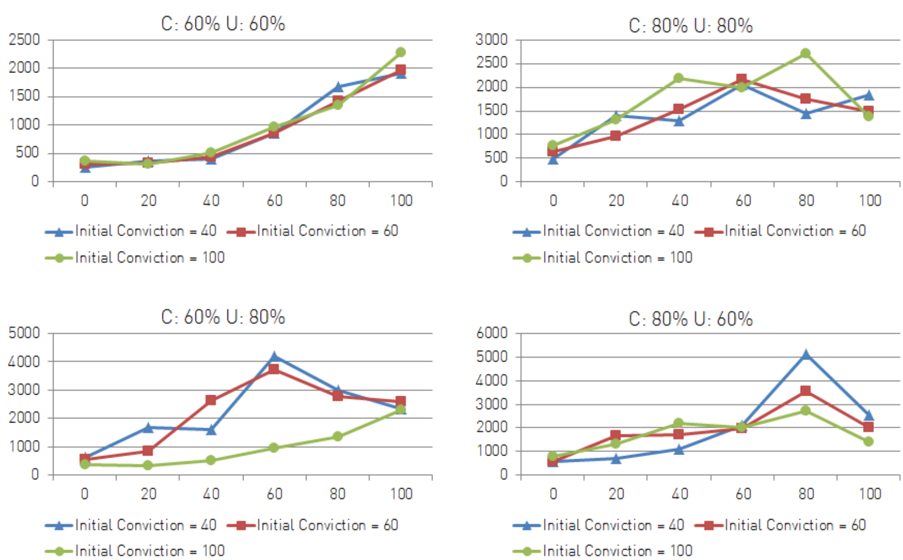


Fig. 6. The time it takes to reach consensus increases drastically with an increase in anonymity level (similarity level = 0.8)

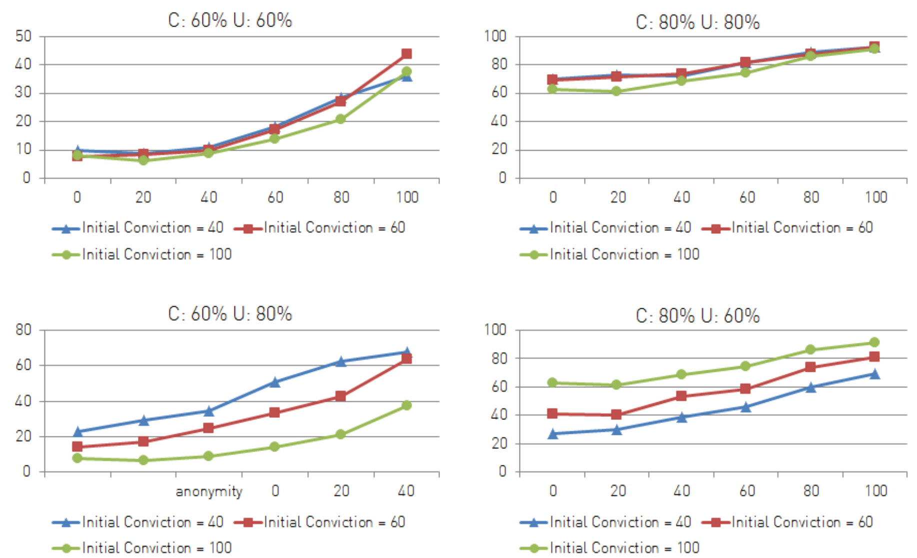


Fig. 7. Number of agents in the biggest cluster increases with an increase in anonymity level (similarity level = 0.8)

6 Discussion

Generally, anonymity brings little difference in an open society where people are willing to communicate even with others who are much different from themselves. In such a society, anonymity could give rise to the negative effect of slowing the opinion diffusion process. In other words, formation of society-wide consensus can be hampered by anonymity, and thus, in an open society, anonymity might be useless, and even harmful.

On the other hand, in a closed society where people tend to communicate with only those who are very similar to themselves, anonymity could facilitate the information flow of the system, and thus increase the social capital. In a closed society, anonymity resulted in greater consensus by embracing more people in a cluster. However, anonymity also increases the time it takes to reach consensus.

We conducted the simulation in the hope that our society is filled more with truth whether or not it is ‘convinced’, and this is reflected in the initial settings. However, if the initial condition is reversed, the result will be the complete opposite. In such a negative situation, anonymity could contaminate the world with falsified opinions, andthere would be difficulty in reaching consensus.

In short, it depends on the context whether anonymity is good or bad. In a society filled with falsified information, anonymity could worsen the situation. False information could be reinforced by more false information from others, and finally falsehood could dominate the whole network. In a society filled more with truth, anonymity could bring more truth to the community, but it takes time. Therefore, we should consider if the time cost is worth taking.

Hofstede [12] identified four basic dimensions that capture characteristics ofdifferentcultures; power distance, uncertainty avoidance, individualism versus collectivism, and masculinity versus femininity. Among them, power distance, uncertainty avoidance, and individualism versus collectivism are related to the similarity threshold. In a society in which collectivism is dominant, someone who has different ideas from social norm will not be tolerated. Furthermore, if people tend to avoid uncertainty strongly, they will turn their back even on those who have only slight differences.Power distance has positive correlation with collectivism. In other words, if a society has strong collectivism, strong uncertainty avoidance, and great power distances,it means that it has high similarity threshold, and thus anonymity will be more effective in that society than in a society with opposite characteristics.

There are some limitations in our study. The number of agents used in the simulation was so small that we cannot say that the results from the simulation reflect our society. Furthermore the simulation was repeated just 30 times for one setting. The result could be distorted by only a single extreme case. Increasing the number of agents and doing more simulations could help generalize the experiment.

In the proposed model, all the opinions are given randomly at first. There's no explanation where the unconvinced opinions come from. The proposed model does not capture those dynamic features of information diffusion, and just draws it as a static picture. In reality, mass media continuously produces new information and spreads. Individuals acquire information from the mass media and they also diffuse it to others. In addition, old information should be removed from the system as old memories are forgotten by people. Sometimes, an opinion is embraced by all the members in a society and becomes a norm. It's immersed into the society and could be redundantor trivial information for a simulation. Including dynamism in the model could help us explain the society more precisely.

In the experiment, we had to assign a value to a parameter to run the simulation, but we couldn't be sure that the parametric values reflected the reality. Without knowing where we are, it's hard to find out where we should go. However, it doesn't mean that the research is useless. At least, we can identify the macroscopic effects of anonymity on the society.

In this research, using a simulation method, we proposed an opinion diffusion model by introducing a conviction dimension and we investigated the way anonymity can affect the society. We have seen that, in general, increasing anonymity also increased the consensus in a society, but it also slowed the process of information flow. We hope that our research model will motivate researchers to further study the anonymity and opinion dynamics to improve our community.

References

1. Afshar, M., Asadpour, M.: Opinion Formation by Informed Agents. Journal of Artificial Societies and Social Simulation 13 (2010)
2. Albert, M.R.: E-Buyer Beware: Why Online Auction Fraud Should Be Regulated. American Business Law Journal 39, 575–644 (2002)
3. Aldashev, G., Carletti, T.: Benefits of Diversity, Communication Costs, and Public Opinion Dynamics. Complexity 15, 54–63 (2009)

4. Axelrod, R.: The Dissemination of Culture: A Model with Local Convergence and Global Polarization. Journal of Conflict Resolution 41, 203–226 (1997)
5. Balcetis, E., Dunning, D.: See What You Want to See: Motivational Influences on Visual Perception. Journal of Personality and Social Psychology 91, 612–625 (2006)
6. Bochner, S., Insko, C.A.: Communicator Discrepancy, Source Credibility, and Opinion Change. Journal of Personality and Social Psychology 4, 614–621 (1966)
7. Brucks, M.: The Effects of Product Class Knowledge on Information Search Behavior. Journal of Consumer Research 12, 1–16 (1985)
8. Christopherson, K.M.: The Positive and Negative Implications of Anonymity in Internet Social Interactions: 'On the Internet, Nobody Knows You're a Dog'. Computers in Human Behavior 23, 3038–3056 (2007)
9. Corritore, C.L., Kracher, B., Wiedenbeck, S.: On-line Trust: Concepts, Evolving Themes, a Model. International Journal of Human-Computer Studies 58, 737–758 (2003)
10. Davenport, D.: Anonymity on the Internet: Why the Price Be Too High. Communications of the ACM 45, 33–36 (2002)
11. Elliston, F.A.: Anonymity and Whistleblowing. Journal of Business Ethics 1, 167–177 (1982)
12. Hofstede, G.: National Cultures in Four Dimensions. International Studies of Man and Organization 13, 46–74 (1983)
13. Laguna, M.F., Abramson, G., Zanette, D.H.: Minorities in a Model for Opinion Formation. Complexity 9, 31–36 (2004)
14. Lea, M., Spears, R., de Groot, D.: Knowing Me, Knowing You: Anonymity Effects on Social Identity Processes within Groups. Personality and Social Psychology Bulletin 27, 526–537 (2001)
15. Lee, E.J.: Deindividuation Effects on Group Polarization in Computer-Mediated Communication: The Role of Group Identification, Public-Self-Awareness, and Perceived Argument Quality. Journal of Communication 57, 385–403 (2007)
16. Mann, L.: The Baiting Crowd in Episodes of Threatened Suicide. Journal of Personality and Social Psychology 41, 703–709 (1981)
17. Postmes, T., Spears, R., Lea, M.: Breaching or Building Social Boundaries?: SIDE-Effects of Computer-Mediated Communication. Communication Research 25, 689–715 (1998)
18. Schlenker, B.R., Leary, M.R.: Social Anxiety and Self-Presentation: A Conceptualization and Model. Psychological Bulletin 92, 641–669 (1982)
19. Suler, J.: The online disinhibition effect. Cyberpsychology& Behavior 7, 321–326 (2004)
20. Teich, A., Frankel, M.S., Kling, R., Lee, Y.: Anonymous Communication Policies for the Internet: Results and Recommendations of the AAAS Conference. The Information Society 15, 71–77 (1999)
21. Weisbuch, G., Deffuant, G., Amblard, F., Nadal, J.P.: Meet, Discuss, and Segregate! Complexity 7, 55–63 (2002)

Citizen Science 2.0: Data Management Principles to Harness the Power of the Crowd

Roman Lukyanenko[1], Jeffrey Parsons[1], and Yolanda Wiersma[2]

[1] Faculty of Business Administration, Memorial University, St. John's Canada
[2] Department of Biology, Memorial University, St. John's Canada
{roman.lukyanenko,jeffreyp,ywiersma}@mun.ca

Abstract. Citizen science refers to voluntary participation by the general public in scientific endeavors. Although citizen science has a long tradition, the rise of online communities and user-generated web content has the potential to greatly expand its scope and contributions. Citizens spread across a large area will collect more information than an individual researcher can. Because citizen scientists tend to make observations about areas they know well, data are likely to be very detailed. Although the potential for engaging citizen scientists is extensive, there are challenges as well. In this paper we consider one such challenge – creating an environment in which non-experts in a scientific domain can provide appropriate and accurate data regarding their observations. We describe the problem in the context of a research project that includes the development of a website to collect citizen-generated data on the distribution of plants and animals in a geographic region. We propose an approach that can improve the quantity and quality of data collected in such projects by organizing data using instance-based data structures. Potential implications of this approach are discussed and plans for future research to validate the design are described.

Keywords: design, citizen science, management, database design, conceptual modeling, data quality.

1 Introduction

Citizen science is a term used to describe the voluntary participation of amateur scientists in scientific endeavors [1]. Humans are increasingly regarded as effective sensors of their environment [2] and the potential for using information collected by individuals is continuously expanding [3]. Citizen science has a long tradition. During the Victorian era many wealthy individuals engaged in natural history as a hobby, and made contributions to the understanding of species distributions and behavior as a result. With the development of the Internet, it has become easier for ordinary people to participate and contribute large amounts of information. Yet, given the expertise and language gap between scientists and ordinary people, information transfer in citizen science projects is not straightforward. While citizen scientists can offer insights and generate new ideas [4], their lack of training and expertise results in inconsistent and incorrect data [5,6,7]. In particular, where direct elicitation of people's opinions is required we can expect lower scientific accuracy of data as wider

H. Jain, A.P. Sinha, and P. Vitharana (Eds.): DESRIST 2011, LNCS 6629, pp. 465–473, 2011.

audiences with lesser expertise get engaged. This research attempts to address this problem by suggesting data management principles that maximize the quantity and quality of information collected from non-experts.

There are many advantages to harnessing citizen scientists. Participants spread across a large area will collect more information than an individual researcher can. Because citizen scientists tend to make observations about areas they know well, data are likely to be very detailed. An additional advantage is the potential longevity of such data; some citizen science programs (e.g., the Audubon Christmas Bird Count) have been in existence for over 100 years, resulting in data sets extending over long periods, thus enabling analysis of trends. Coupled with the availability of relatively inexpensive photo and video equipment, harnessing the power of ordinary people to provide data and observations about the natural world can lead to major advances in the natural sciences, as well as assist in vital areas of wildlife conservation and emergency management in the event of natural disasters (such as the Gulf of Mexico oil spill).

Although the potential for engaging citizen scientists is extensive, there are challenges as well. In this paper we describe one such challenge – creating an online environment in which non-experts in a scientific domain can provide appropriate and accurate data regarding their observations. We describe the problem in the context of a research project that includes the development of a website and database to collect citizen-generated data on the distribution of plants and animals in a geographic region. We propose an approach to improving the quantity and quality of data collected in such projects by using instance-based data structures [8]. Potential implications of this approach are discussed and plans for future research to validate the design are described.

2 The Challenge – Facilitating Participation

The success of a citizen science project depends on the willingness and ability of members of the general public to voluntarily observe and report information. In many cases, this in turn requires some level of scientific knowledge by participants. For example, the website of the Cornell Ornithology Lab, eBird (www.ebird.com), draws on the enthusiasm of avid birders to provide detailed information about bird sightings. The Cornell Lab is an international leader in ornithological research, and eBird is an exemplar of a successful online citizen science project. However, engagement of the lay public with eBird may be limited by the application domain. Citizen scientists who wish to upload bird sightings need to be familiar with bird taxonomy and identification. The bird checklist provided in the online interface assumes the user has already made a positive identification (i.e., identified the species) and knows to which taxonomic group the bird belongs. This is acceptable for a reasonably experienced citizen scientist, but the rank beginner ([7] provides a taxonomy of “expertise levels” among citizen scientists) may not be able to participate, or may provide data of poor quality as a result of his/her inability to make a positive identification [9]. Thus, useful participation may be limited to more experienced amateurs.

The issue of quality and reliability of user-supplied data in citizen science projects has attracted much attention in recent research [5]. Although the literature is limited (given the relative recency of Web 2.0 applications), the implied assumption of much of the work to date is that there exists an inherent trade-off between data quality and the level of participation (data quantity). Experts are considered to be the source of the most accurate volunteered information [7], but there are fewer “expert amateurs” than “beginners” available to participate.

The common method of increasing data quality considered in the literature is training and educating the volunteers. For example, data inconsistency may result from volunteers’ lack of experience, inadequate guidelines and insufficient training [4], “rolled up” into larger monitoring projects [6]. Training, while generally desirable, may not always be possible, especially for low budget projects.

A typical way to increase quality is through expert verification, an approach that has been used for by-catch and beached bird observation [12] and for unusual observations on eBird [19]. However, with the size of data sets increasing [6], individual verification becomes unrealistic and in many ways is contrary to the spirit of citizen science.

Another line of research suggests social networking as key to increasing data quality. Some research has proposed a trust and reputation model for classifying knowledge using the social networking practice of peer evaluation of content [13]. This approach is the basis for iSpot, a website that exploits a user reputation mechanism to determine accuracy of observations [14]. The reputation-trust model adapted from well-developed trust research in e-commerce [e.g., 15,16,17] has been applied to the context of citizen science [18]. While the social networking approach appears promising, it has a number of limitations. Although it has been compared to the “scientific peer review process” [13], social networking is useful only for popular citizen science projects with large numbers of users. Web sites with a small number of users may not have sufficient user activity per observation to ensure rigorous peer review. In addition, as even a very popular website cannot guarantee that every observation will receive equal scrutiny, this metaphor of scientific review is not fully justified. Furthermore, users with high reputation who are considered experts in some domain may still provide inaccurate data in other domains. Most importantly, social networking may fail to harness the potential of an individual non-expert, as in the absence of domain knowledge such volunteers may feel too intimidated to express their opinion (consider the description of a type 'neophyte' [7]). Finally, the social networking approach lacks generality, as it relies on a particular technology, and may exclude many citizen science projects that do not currently employ a social networking model.

Notwithstanding the value of the above approaches, we argue that it is possible to increase the quality of data generated by of an individual volunteer by minimizing subject information that has a high likelihood of being inaccurate. Requiring volunteers to make a (potentially inaccurate) positive identification of natural history phenomena implies that the observer has some knowledge of traditional scientific taxonomy. We argue that an alternative to classifying observations according to a fixed taxonomy is to allow volunteers to provide information about observations and that this will increase the general success of citizen-scientist projects.

3 A Proposed Solution – Attribute-Based Data Collection

A traditional approach to citizen participation in scientific data collection works well (i.e., makes it possible to collect accurate data from a broad constituency) only if the participants are capable of classifying observed phenomena accurately. For example, accurate classification of observed plants and animals by species requires that participants understand the distinguishing characteristics of species. We contend that imposing this requirement on participation, as in projects such as eBird, imposes a severe and unnecessary restriction on the level of participation that can be realized in citizen science projects.

To combat this limitation, we propose an approach to data collection and storage that does not require users to classify observed phenomena. Instead, they record any attributes associated with the observation. We illustrate the approach in the context of NLNature – an ongoing citizen scientist-based project to collect data about the flora and fauna of Newfoundland & Labrador (www.nlnature.com). Our proposal is based on the instance-based data model (IBDM) [8] and our application of the model has implications both for interface design and for database design. Working within the framework of the IBDM, we extend the model to address issues of identifying phenomena, and suggest how the model offers a solution to the challenges of a typical citizen scientist project.

The IBDM is based on ontological and cognitive principles [8, 20]. Ontologically, every "thing" possesses a unique set of properties. Classes are formed based on the principle that one can classify things based on a subset of their observed properties, and make inferences about unobserved properties the instance possesses by virtue of belonging to the class [21]. Since an instance can possess very many properties, it can belong to a very large number of potential classes, depending on the context.

By shifting the focus from a predefined classification to the thing (instance) and its attributes (see Fig. 1) we do not need to model a domain *a priori* in terms of the classes of interest. It is sufficient to ensure that the application has a comprehensive collection of instances, and each instance contains a set of well-defined attributes. When required, a user can assemble a dynamic classification based on the collection of attributes that are of interest at a given moment. For example, if an attribute such as "behavior" is of interest, then at least two classes can be constructed based on values: animals that are nocturnal (active at night) vs. diurnal (active during the day). The same system can also use attributes that connect each species with a biological taxonomy to reproduce scientific biological classification. Thus, the instance-based model is capable of achieving the objectives of a traditional classification without the inherit limitations.

We posit that attribute-based design will enable potential citizen scientists to provide data efficiently and effectively, thereby increasing their participation in data gathering. We propose a data collection interface designed based on the primacy of a phenomenon and its attributes over classification of the phenomenon. A user is asked to identify those attributes (e.g., size, color, appearance, behavior, location, sound) of an observed plant or animal. In principle, the primary scientific object of an observation (the species observed) can be identified by an expert after the observation is recorded, provided that the user reports enough attributes to produce a positive

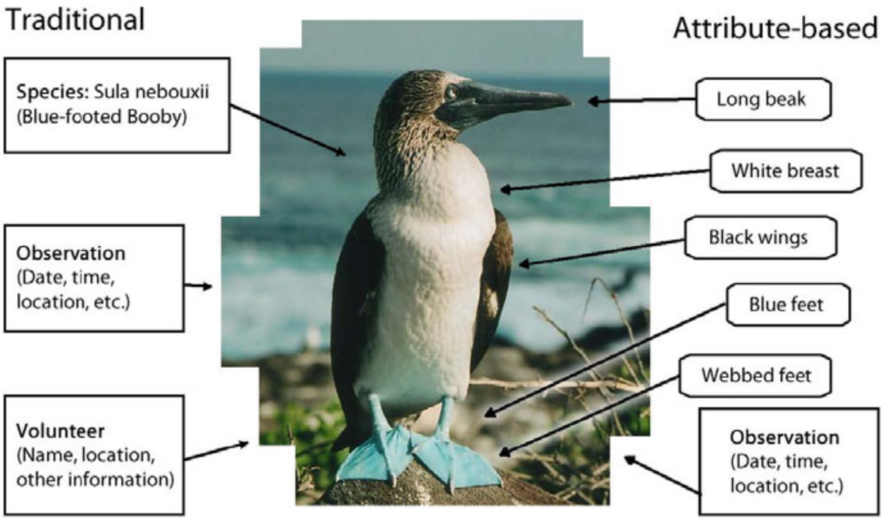


Fig. 1. Traditional vs. attribute-based information (Image source: Wikimedia Commons)

identification. This contrasts with traditional approaches requiring *a priori* classification (e.g., requiring users to select from a checklist of species), which are usable only by more expert volunteers. Once several attributes are selected, the system will match them with pre-existing sets of identifying attributes for species, and either infer a species or ask for additional attributes that could also be automatically inferred from those previously supplied.

Although the final attribute set resulting from an observation can potentially match multiple species, this proposed solution offers a realistic compromise. Non-experts do not always know the phenomenon that was observed. It is more realistic to expect a volunteer to remember some features of unknown species then to expect a precise classification and identification. The key activity of identification therefore shifts from designing a perfect classification to facilitating effective attribute management. The more the system can guide the choice of attributes, the higher inferential value such records hold, and the easier it is to classify observations.

4 Attribute-Based Database Design

Database structure can be either a major inhibitor or a facilitator of system evolution [8, 20, 22]. Traditionally, database design results in a representation of the application domain as a set of related classes (translated to tables in a relational database). In addition, once the database structure is established it is assumed to be relatively static, allowing other application elements, such as program code, to be created based on the static structure. Altering the database structure once a system is built is costly. Thus, traditional database design is subject to the inherit limitations of a rigid classification [8].

The collection of user-supplied information based on attributes of observations suggests the need for a database structure that supports variability in the data collected from observers, including failure to classify an observation. Support for flexible attribute collection can be implemented using a traditional relational database, as illustrated in Fig. 2. We propose storing attributes in a generic table "Attributes" that contains attribute name and a unique identifier. A separate "Attributes-Relationships" table links one attribute to another and creates relationships between attributes. The table contains the primary key from the parent attribute and a primary key from the child attribute, thus making many-to-many relationships possible. For example, if the user selects the attribute "was flying" then "lives in water" will be automatically removed from the interface, and the system will respond by presenting a new set of potential attributes that can be inferred from "was flying" (e.g., "has feathers", "seen at night", "six legs"). The choice of the first attribute narrows the observation to a bird (subsequent attributes could focus on feather color, beak size, habitat, etc.), the second to a bat, and the third to a flying insect.

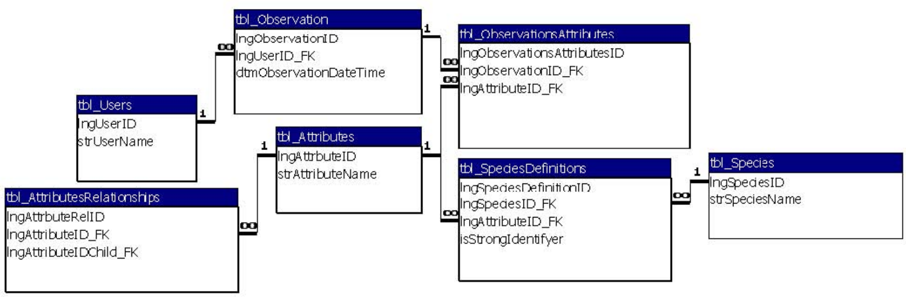


Fig. 2. ER diagram showing instance-based data structure for a typical citizen scientist project

In order to match selected attributes against a class-defining set, class blueprints for each species need to be maintained. This is achieved by a table of "Species Definitions" that links species with their attributes via a one-to-many relationship. For example, boreal felt lichen will link to the following attributes: fuzzy white fringe around the edges, grayish-brown when dry, has red dots, leafy, slate-blue when moist. User-observed properties are then matched against the class definitions to infer class membership. If necessary, new class definitions can be added or existing ones altered during the operational phase of the enterprise system without having to change the database schema. Finally, we provide tables that join objects and attributes to store the details of the observations. These tables store events in the system. Each table includes primary keys from the attributes and objects tables, attribute values, and date/time of the attribute creation/change. By recording the date of attribute creation/change, the system can document events that happen to the same phenomenon. This approach addresses a persistent issue of database design – adaptation to organizational change – that a traditional approach with its reliance on rigid classification struggles to resolve [8, 22].

5 Implications for Data Gathering

The attribute based system proposed for this citizen scientist project has the potential to increase participation rates (and, hence, data quantity). Unlike natural history websites that only present taxonomic checklists and assume a basic level of expertise from citizen scientists, the system proposed here allows for the full spectrum of volunteer contributors [7] to participate. We believe that this will provide a means of validating user-supplied data within the user community, particularly if users supply additional information with their observations (e.g., photographs) that can be reviewed by experts when necessary.

Many citizen science projects provide inventory data across space and time. Although there will be biases within the data (for example, to areas where there is high human population density and to more charismatic or easily observable species), the data do have the benefit of indicating long-term trends. For the scientific community, the biggest value is that such data sets are generated by many "eyes on the ground;" thus, there is a higher likelihood of rare or unusual species being detected or for early detection of new trends. Hence, it is important to have a usable system that promotes a broad and consistent level of participation. Some potential uses of data collected this way might be unanticipated. For example, long term data can be useful to identify benchmark conditions in the event of a natural or anthropogenic disaster (e.g., the Gulf oil spill) and can guide restoration strategies.

This research explores general ways of facilitating information transfer between users with different level of domain expertise within the context of a citizen science project. Information systems are increasingly being used to collect data from ordinary people (e.g. personal health records [23]). While a number of factors are considered to influence information quality (e.g., [24]), little attention is given to the role of data structures in ensuring quality of collected information.

6 Limitations and Future Research

Internet technologies open new opportunities for citizen science. Yet the knowledge requirements implied by rigid data structures constrain effective participation of novices and thereby limit the potential outreach of citizen science projects. A successful implementation of the approach proposed in this paper can facilitate development of citizen-scientist initiatives. We believe it also has broader applications based on user-generated content, and promises to be a practical solution to an important design problem in citizen science.

The foundation of our proposed approach to improving the quantity and quality of citizen science projects is the IBDM [8]. The primary theoretical assumption of the IBDM – that existence of *things* and *properties* (attributes) precedes classification - has generally [cf. 25] been supported in ontological [26,27] and cognitive research [28]. However, while attributes are building blocks of classification [29], not all classes can be *efficiently* expressed as sets of common attributes (e.g., radial categories [30,31,32]). Moreover, many superordinate categories, such as *furniture, animal, vehicles* tend to be abstract and reflect some rules or functions rather than observable attributes [33-34]. While this appears to limit the scope of our model, we believe that for practical reasons little information in citizen science projects will be expressed in terms of higher-level categories. Indeed, humans prefer to avoid superordinate categories when they think of individual objects [35].

Classification is a ubiquitous activity and an attribute-centered approach to knowledge management needs to be tested to determine its technological, economic, scientific and business utility. We are currently designing empirical studies to measure the practical impact of the above approach on data collection and storage, user participation and satisfaction, data quality, and usefulness to scientists. The experiment will also test the overall effectiveness and feasibility of applying the IBDM to empower citizen scientists.

References

1. Silvertown, J.: A new dawn for citizen science. Trends in Ecology and Evolution 24, 467–471 (2009)
2. Goodchild, M.: Citizens as sensors: the world of volunteered geography. GeoJournal 69, 211–221 (2007)
3. Hand, E.: People power. Nature 466, 685–687 (2010)
4. Foster-Smith, J., Evans, S.M.: The value of marine ecological data collected by volunteers. Biological Conservation 113, 199–213 (2003)
5. Flanagin, A., Metzger, M.: The credibility of volunteered geographic information. GeoJournal 72, 137–148 (2008)
6. Wiersma, Y.F.: Birding 2.0: citizen science and effective monitoring in the Web 2.0 world. Avian Conservation and Ecology 5, 13 (2010)
7. Coleman, D.J., Georgiadou, Y., Labonte, J.: Volunteered geographic information: The nature and motivation of producers. International Journal of Spatial Data Infrastructures Research 4, 332–358 (2009)
8. Parsons, J., Wand, Y.: Emancipating instances from the tyranny of classes in information modeling. ACM Transactions on Database Systems 25, 228 (2000)
9. Parsons, J., Lukyanenko, R., Wiersma, Y.: Easier citizen science is better. Nature 471, 37 (2011)
10. Dickinson, J.L., Zuckerberg, B., Bonter, D.N.: Citizen science as an ecological research tool: challenges and benefits. Annual Review of Ecology, Evolution, and Systematics 41, 112–149 (2010)
11. Aaron, W.E.G., Tudor, M.T., Haegen, W.M.V.: The Reliability of Citizen Science: A Case Study of Oregon White Oak Stand Surveys. Wildlife Society Bulletin 34, 1425–1429 (2006)
12. Hamel, N.J., Burger, A.E., Charleton, K., Davidson, P., Lee, S., Bertram, D.F., Parrish, J.K.: Bycatch and beached birds: Assessing mortality impacts in coastal net fisheries using marine bird strandings. Marine Ornithology (2009)
13. Bishr, M., Mantelas, L.: A trust and reputation model for filtering and classifying knowledge about urban growth. GeoJournal 72, 229–237 (2008)
14. Silvertown, J.: Taxonomy: include social networking. Nature 467, 788–788 (2010)
15. Komiak, S.Y.X., Benbasat, I.: The effects of personalization and familiarity on trust and adoption of recommendation agents. MIS Quarterly 30, 941–960 (2006)
16. Gefen, D., Karahanna, E., Straub, D.W.: Trust and TAM in online shopping: An integrated model. MIS Quarterly 27, 51–90 (2003)

17. Palvia, P.: The role of trust in e-commerce relational exchange: A unified model. Information & Management 46, 213–220 (2009)
18. Alabri, A., Hunter, J.: Enhancing the quality and trust of citizen science data. In: IEEE eScience 2010, Brisbane, Australia (2010)
19. Sullivan, B.L., Wood, C.L., Iliff, M.J., Bonney, R.E., Fink, D., Kelling, S.: eBird: A citizen-based bird observation network in the biological sciences. Biological Conservation 142, 2282–2292 (2009)
20. Parsons, J., Su, J.: Analysis of data structures to support the instance-based data model. In: International Conference on Design Science Research in Information Systems and Technology (DESRIST), pp. 107–130 (2006)
21. Parsons, J., Wand, Y.: A question of class. Nature 455, 1040–1041 (2008)
22. Allen, B.R., Boynton, A.C.: Information architecture: In search of efficient flexibility. MIS Quarterly 15, 435–445 (1991)
23. Agarwal, R., Angst, C.M.: Technology-Enabled Transformations in U.S. Health Care: Early Findings on Personal Health Records and Individual Use. In: Galletta, D., Zhang, P. (eds.) Human-Computer Interaction and Management Information Systems: Applications. M.E. Sharpe, Inc., Armonk (2006)
24. Nicolaou, A.I., McKnight, D.H.: Perceived information quality in data exchanges: Effects on risk, trust, and intention to use. Information Systems Research 17, 332–351 (2006)
25. Grill-Spector, K., Kanwisher, N.: Visual recognition. Psychological Science 16, 152–160 (2005)
26. Bunge, M.A.: Treatise on Basic Philosophy: The furniture of the world. Reidel, Dordrecht (1977)
27. Wand, Y., Weber, R.: An ontological model of an information system. IEEE Transactions on Software Engineering 16, 1282–1292 (1990)
28. Bowers, J.S., Jones, K.W.: Detecting objects is easier than categorizing them. Quarterly Journal of Experimental Psychology 61, 552–557 (2008)
29. Wand, Y., Monarchi, D.E., Parsons, J., Woo, C.C.: Theoretical foundations for conceptual modeling in information systems development. Decision Support Systems 15, 285–304 (1995)
30. Raccoon, L.S.B., Puppydog, P.O.P.: A middle-out concept of hierarchy (or the problem of feeding the animals). SIGSOFT Softw. Eng. Notes 23, 111–119 (1998)
31. Young, J.J., Williams, P.F.: Sorting and comparing: Standard-setting and "ethical" categories. Critical Perspectives on Accounting 21, 509–521 (2010)
32. Lakoff, G.: Women, fire, and dangerous things: what categories reveal about the mind. University of Chicago Press, Chicago (1987)
33. Rosch, E., Mervis, C.B., Gray, W.D., Johnson, D.M., Boyesbraem, P.: Basic objects in natural categories. Cognitive Psychology 8, 382–439 (1976)
34. Murphy, G.L., Wisniewski, E.J.: Categorizing objects in isolation and in scenes - What a superordinate is good for. Journal of Experimental Psychology: Learning 15, 572–586 (1989)
35. Rorissa, A.: User-generated descriptions of individual images versus labels of groups of images: A comparison using basic level theory. Information Processing & Management 44, 1741–1753 (2008)

Author Index